ilmedia

Technische Universität Ilmenau

Hildenbrandt, Regina:

The k-server problem with parallel requests and the compound work function algorithm
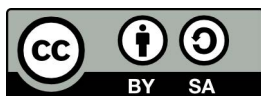
The k-Server Problem with Parallel Requests and the Compound Work Function Algorithm

R. HILDENBRANDT

Institute of Mathematics, Ilmenau Technical University,
PF 10 06 65, 98684 Ilmenau, Germany

`r.hildenbrandt@tu-ilmenau.de`

Abstract. In this paper the compound work function algorithm for solving the generalized k-server problem is proposed. This problem is an online k-server problem with parallel requests where several servers can also be located on one point. In 1995 Koutsoupias and Papadimitriou have proved that the well-known work function algorithm is competitive for the (usual) k-server problem. A proof, where a potential-like function argument is included, was given by Borodin and El-Yaniv in 1998. Unfortunately, certain techniques of these proofs cannot be applied to show that a natural generalization of the work function algorithm is competitive for the problem with parallel requests. Values of work functions, which are used by the compound work function algorithm are derived from a surrogate problem, where at most one server must be moved in servicing the request in each step. We can show that the compound work function algorithm is competitive with the same bound of the ratio as in the case of the usual problem.

1   Introduction

This paper deals with a generalized online k-server problem. We want to present a new algorithm, the "compound work function algorithm", and show its qualities for solving this problem.

In Hildenbrandt (2014) we have introduced a generalized k-server problem with parallel requests where several servers can also be located on one point. The investigation of the generalized k-server problem was initiated by an operations research problem which consists of optimal conversions of machines or moulds (see Hildenbrandt (1993)). We are given initial locations of k servers in a metric space. Requests R^t for service at several points come in over time. It is sensible in the case of such requests to

distinguish the surplus-situation where the request can be completely fulfilled by means of the k servers and the scarcity-situation where the request cannot be completely met. Immediately after the t-th request is received, a sufficient number of servers must be moved from theirs current locations to the request points. The choice of which servers are moved, in the case of the surplus-situation or the choice of requests which should be fulfilled, in case of the scarcity-situation, respectively, must be made based only on the current servers configuration and on the requests seen so far; that is, the requests R^u for $u \leq t$. Moving servers costs the distances the servers are moved, and the goal is to minimize the total cost. Firstly, we have proposed the (compound) Harmonic k-server algorithm in the cases of unit and general distances, respectively (see Hildenbrandt (2014 and 2016)) and proved the competitiveness of these algorithms. The main subject of this paper are work function algorithms.

In online computation, an algorithm must decide how to act on incoming requests without any knowledge of future inputs. In contrast, an offline procedure would be allowed to know the entire sequence of requests in advance, before it makes any decisions. We want online algorithms whose cost compares favorably to the cost of an optimal offline algorithm. A deterministic online algorithm is c-competitive if there exists a function α of the initial configuration so that for every finite input sequence the cost incurred by the algorithm is bounded by α plus c times the minimum cost of processing the input sequence (see Sleator and Tarjan (1985)). c is called the competitive ratio of the algorithm. The competitive ratio is a measure of how much better we could do if we knew the future.

The (usual) k-server problem (where at most one server must be moved in servicing the request in each step) was introduced by Manasse, McGeoch and Sleator (1988, 1990). Meanwhile it is the most studied problem in the area of competitive online problems. Historical notes on k-server problems can be found in the book by A. Borodin and R. El-Yaniv (1998) (Sections 10.9 and 11.7) or also in the paper by Y. Bartal and E. Grove (2000). There two important results are "competitiveness" of the deterministic work-function algorithm (see E. Koutsoupias and C. Papadimitriou (1995)) and of the randomized Harmonic k-server algorithm against an adaptive online adversary (see Y. Bartal and E. Grove (2000)). The idea for an online work function algorithm was implicitly introduced by Borodin, Linial and Saks (1992) in connection to "metrical task systems". A work function algorithm for the (usual) k-server problem has been proposed by several researchers (for more detail, see E. Koutsoupias and C. Papadimitriou (1995), p. 974). The competitiveness of the work function algorithm in any metric space was proved by Koutsoupias and Papadimitriou (1995). In the present paper we use the proof by Borodin and El-Yaniv (1998) (see Theorem 10.9 and its proof). This proof is based on Koutsoupias's dissertation (1994) and on a presentation of this proof due to Bartal that offers a simpler potential-like function argument (see A. Borodin and R. El-Yaniv (1998), pp. 178, 179). There are also existing k-server problems which, on the one hand, are generalizations of the usual problem and otherwise include specific properties. Such problems and corresponding work function algorithms was described

by R. Sitters (2013) (for example).

Unfortunately, certain techniques of the above-mentioned proofs from literature cannot be applied to show that a natural generalization of the (usual) work function algorithm (see Section 4.) is competitive for the problem with parallel requests. Therefore we will develop the "compound work function algorithm" for the generalized k-server problem (in the case of the surplus-situation) in Section 5. Values of work functions, which are used by the compound work function algorithm are derived from a surrogate problem, where at most one server must be moved in servicing the request in each step. We will prove that the compound work function algorithm is competitive with the same bound of the ratio as in the case of the usual problem. In order to prove the competitiveness we also need the optimal solution of the corresponding offline problem. For this we introduce generalized work functions in Section 3.

2 The formulation of the model

Now, we want to describe the usual k-server problem (a)) and the generalized problem with parallel requests (b)).[1] We are given initial locations of k servers in a metric space (M, d). Several servers can be located on one point (b)). Requests

a) r^t

b) R^t

for service

a) at points $\{p_t\}$

b) at finite multisets of points $\{P_t\}$ ("parallel requests")

come in over time. Let

a) $\sigma = r^1, r^2, \cdots, r^n$

b) $\sigma = R^1, R^2, \cdots, R^n$

such a sequence of requests. Immediately after the t-th request is received,

a) one of the servers must be moved from its current location to p_t.

b) a sufficient number of servers must be moved from theirs current locations to the points of the multiset P_t. The choice of which

a) server is moved

b) servers are moved, in the case of the surplus-situation or the choice of requests which should be fulfilled (by means of the k servers), in case of the scarcity-situation, respectively (in more detail see below)

must be made based only on the current servers configuration and on the requests seen so far; that is, the requests

a) r^t for $u \leq t$.

b) R^u for $u \leq t$.

Moving servers costs the distances the servers are moved.

Our goal is to find a deterministic online algorithm for solving the generalized k-server problem (b)) with the same bound of the ratio as the well-known work function algo-

---

[1] See also Hildenbrandt (2014 and 2016).

rithm (for solving the usual problem (a)), where the bound is $2k - 1$.

A server configuration of the algorithm is viewed in this paper as a multiset of k (not necessarily distinct) points representing the locations of the servers. A request can also be viewed as a multiset, but with a arbitrary number of (not necessarily distinct) points.

We denote multisets by capital letters and points of M by small letters. For any (finite) multiset C with points from M we use the notation $C = \{p_1, p_2, ..., p_m\}$ where p_j, $j = 1, 2, \cdots, m$ are (not necessarily distinct) points from M.

For multisets C_1 and C_2 we let $C_1 + C_2$ denote the multiset union; similarly, $C_1 - C_2$ denotes multiset exclusion. For a point p and configuration C, we abbreviate $C + \{p\}$ by $C + p$, $C - \{p\}$ by $C - p$, $C + \{p, p\}$ by $C + 2p$ and so on. So, for example, if $p \in C$ then the configuration $C + p$ contains at least two copies of p and $C - p + p = C + p - p$ contains p. However, if $p \notin C$ then $C - p = C$ and $C - p + p \neq C + p - p$. (See A. Borodin, R. El-Yaniv (1998), p. 164.) Let $C(p), p \in C$ be the subset of C with $p \notin C - C(p)$. We set $C(p) = \emptyset$ if $p \notin C$.

Finally, $|C|$ denotes the number of elements of a finite multiset C.
For every two finite multisets X and Y with elements from M and $|X| = |Y|$, we define $D(X, Y)$ the set distance as the value of the minimum weight matching between X and Y (see Appendix A.).

Then the inequality $|R| \leq k$ describes the surplus-situation. The request R can be completely fulfilled. We say a request $R = \{p_1, p_2, ..., p_m\}$ is served if at least $|R(p)|$ servers are at point $p, p \in M$. In contrast, $|R| > k$ means the scarcity-situation. The request cannot be completely met, however it should be met as much as possible. The request R is served if at most $|R(p)|$ servers are at point $p, p \in M$.

For any request sequence σ and any k-server algorithm ALG, $\text{ALG}(\sigma)$ is defined as the total distance moved by the ALG's servers in servicing σ.

Analogous to A. Borodin, R. El-Yaniv (1998), p. 152 working with lazy algorithms ALG is sufficient. This means, servers are not moved in a step if they are not needed to fulfil requests in this step. For that reason we define the set of feasible servers' configurations with respect to the previous servers configuration S and the request R in the following way

$$\hat{A}_k(S, R) = \left\{ \bar{S} \in S_{(k)} \,\middle|\, |R(p)| \leq |\bar{S}(p)| \leq \max\{|S(p)|, |R(p)|\}, \; p \in M \; \right\}, \quad (1)$$

where

$$S_{(k)} = \left\{ \bar{S} \mid |\bar{S}| = k \right\}, \tag{2}$$

in the case of the surplus-situation and

$$\hat{A}_k(S, R) = \left\{ \bar{S} \in S_{(k)} \,\middle|\, \min\{|S(p)|, |R(p)|\} \leq |\bar{S}(p)| \leq |R(p)|, \; p \in M \right\} \tag{1a}$$

in the case of the scarcity-situation.

The metric d implies that $(S_{(k)}, D)$ is also a metric space (see R. Hildenbrandt (1995), Lemma 3.6)[2].

3 Generalized work functions and the solution of the offline problem

Next we want to generalize the definition of the work functions, which are used in the work function algorithm applied to the usual k-sever problem (see A. Borodin and R. El-Yaniv (1998), p. 164, 165 or S. O. Krumke and J. Rambau (2005), p. 74, 75 for example). We will use this generalization to solve the problem with parallel request.

Fix a request sequence, $\sigma = R^1, R^2, \cdots, R^n$, and use σ_t to denote the prefix, $R^1, R^2, \cdots, R^t$, of σ. Use $\emptyset$ to denote the empty request sequence. Work functions are defined in terms of an initial configuration S^0 and a request sequence. Specifically, for each configuration S^0, configuration S, the k-server work function $w_t(S) = w_{\sigma_t}(S) = w_{\sigma_t}(S^0, S)$ is defined as the optimal offline cost (sequentially) servicing all requests in σ_t, starting from the initial configuration S^0 and ending at configuration S. Notice that the configuration S does not necessarily contain the last request R^t of σ_t (or any requested points in σ_t). (Thus, $w_t : S_{(k)} \to \mathbb{R}_+$ for given S^0 and σ_t.)

Work functions can be computed recursively as follows. For each configuration S, the initial work function $w_\emptyset(S)$ is simply the configuration distance between S^0 and S. That is

$$w_\emptyset(S) = D(S^0, S). \tag{3}$$

Assume that the value $w_t(S)$ is known for any configuration S. Given the next request R^{t+1} and a configuration S, the value of $w_{t+1}(S)(= w_{\sigma_t, R^{t+1}}(S))$ is computed as follows. If $R^{t+1} \subseteq S$ (surplus-situation) or $S \subseteq R^{t+1}$ (scarcity-situation), respectively, then clearly $w_{t+1}(S) = w_t(S)$. Otherwise, the optimal offline algorithm that must first serve the request R^{t+1} (as much as possible in the case of the scarcity-situation) before ending up in configuration S can first process the sequence σ_t, R^{t+1} ending up in some configuration $\tilde{S} \supseteq R^{t+1}$ (surplus-situation) or $\tilde{S} \subseteq R^{t+1}$ (scarcity-situation), respectively and then move to configuration S. Thus, it follows

$$
\begin{aligned}
w_{t+1}(S) &= \min_{\tilde{S}:\begin{cases} \tilde{S} \supseteq R^{t+1} \text{ if } |R| \le k \\ \tilde{S} \subseteq R^{t+1} \text{ if } |R| > k \end{cases}} \{w_{t+1}(\tilde{S}) + D(\tilde{S}, S)\} \\
&= \min_{\tilde{S}:\begin{cases} \tilde{S} \supseteq R^{t+1} \text{ if } |R| \le k \\ \tilde{S} \subseteq R^{t+1} \text{ if } |R| > k \end{cases}} \{w_t(\tilde{S}) + D(\tilde{S}, S)\}.
\end{aligned}
\tag{4}
$$

The second equality in the above equations is due to the fact that $\tilde{S} \supseteq R^{t+1}$ (surplus-situation) or $\tilde{S} \subseteq R^{t+1}$ (scarcity-situation), respectively. (3) and

---

[2] The proof of Lemma 3.6 remains valid in case of an inifinite metric space.

(4) provides the means for computing the work functions.

From the definition of the work functions follows the property

$$w_\emptyset(A) \leq w_\emptyset(B) + D(A, B) \text{ and}$$
$$w_t(A) \leq w_t(B) + D(A, B) \text{ for } t = 1, 2, \cdots, n \tag{5}$$

for any server configurations A and B.

The proof of (5) is simple. If $w_t(B) = w_t(\tilde{B}^*) + D(\tilde{B}^*, B)$, where $\tilde{B}^*$ is an optimal solution of (4), then also $w_t(A) \leq w_t(\tilde{B}^*) + D(\tilde{B}^*, A)$. Subtracting the equation from the inequality, together with the triangle-inequality $D(\tilde{B}^*, A) - D(\tilde{B}^*, B) \leq D(A; B)$ we get (5). $\qquad\square$

Using (5) and the triangle-inequality, (4) can be simplified a little:

In case of the *surplus-situation* it is sufficient to consider $\tilde{S} = S - X + R^{t+1}$ which differs from S by at most $|R^{t+1} - S|$ points. Then, we can compute the work functions as

$$w_{t+1}(S) = \min_{X \subseteq S, |X| = |R^{t+1}|} \{w_{t+1}(S - X + R^{t+1}) + D(R^{t+1}, X)\}$$
$$= \min_{X \subseteq S, |X| = |R^{t+1}|} \{w_t(S - X + R^{t+1}) + D(R^{t+1}, X)\} \tag{6}$$

or as

$$w_{t+1}(S) = \min_{\hat{X} \subseteq \hat{S}, \, |\hat{X}| = |\hat{R}^{t+1}|} \{w_{t+1}(S - \hat{X} + \hat{R}^{t+1}) + D(\hat{R}^{t+1}, \hat{X})\}$$
$$= \min_{\hat{X} \subseteq \hat{S}, \, |\hat{X}| = |\hat{R}^{t+1}|} \{w_t(S - \hat{X} + \hat{R}^{t+1}) + D(\hat{R}^{t+1}, \hat{X})\}, \tag{7}$$

where

$$\hat{R}^{t+1} := R^{t+1} - S, \; \hat{S} := S - R^{t+1}. \tag{8}$$

If X is a solution of (6) then there exists a solution $\hat{X}$ of (7), and vice versa, so that

$$X = \hat{X} + \tilde{X} \text{ with } \tilde{X} = S \cap R^{t+1}. \tag{9}$$

If we set $\tilde{S} = X$ then (4) has the representation:

$$w_{t+1}(S) = \min_{X \subseteq R^{t+1}, |X| = k} \{w_{t+1}(X) + D(S, X)\}$$
$$= \min_{X \subseteq R^{t+1}, |X| = k} \{w_t(X) + D(S, X)\} \tag{10}$$

in the case of the *scarcity-situation*. However, it is sufficient to consider $\tilde{S} = \hat{X} + (S \cap R^{t+1})$ which differs from S by at most $|S - R^{t+1}|$ points. Then it follows:

$$w_{t+1}(S) = \min_{\hat{X} \subseteq \hat{R}^{t+1}, |\hat{X}| = |\hat{S}|} \{w_{t+1}(\hat{X} + (S \cap R^{t+1})) + D(\hat{S}, \hat{X})\}$$
$$= \min_{\hat{X} \subseteq \hat{R}^{t+1}, |\hat{X}| = |\hat{S}|} \{w_t(\hat{X} + (S \cap R^{t+1})) + D(\hat{S}, \hat{X})\}, \tag{11}$$

with $\hat{R}^{t+1}$ and $\hat{S}$ as above.

If X is a solution of (10) then there exists a solution $\hat{X}$ of (11), and vice versa, so that

$$X = \hat{X} + \tilde{X}, \text{ with } \tilde{X} = S \cap R^{t+1}. \tag{12}$$

Given any initial configuration and any request sequence σ, we can use work functions to compute the optimal offline cost to serve σ, which is

$$OPT(\sigma) = \min_S w_\sigma(S). \tag{13}$$

Obviously, each different point of an optimal configuration S is an element of $\bar{M} \subseteq M$, where

$$p \in \bar{M} \text{ if and only if } p \in S^0 \text{ or } p \in R^t, t \in \{1, 2, \cdots, n\}. \tag{14}$$

Remark 1 *Let S be any configuration.*

Work functions can be computed by means of dynamic programming, which proceeds backward in time. This leads to a sequence of configurations

$$S =: S^{t+1}, \quad S^t, \quad \cdots, \quad S^2, \quad S^1, \quad (S^0)$$

and a sequence of multisets

$$X_t^*, \quad X_{t-1}^*, \quad \cdots, \quad X_2^*, \quad X_1^*,$$

with $S^{\bar{t}} = S^{\bar{t}+1} - X_{\bar{t}}^* + R^{\bar{t}} \supseteq R^{\bar{t}}$ and $|X_{\bar{t}}^*| = |R^{\bar{t}}|$ in case of the surplus-situation or $S^{\bar{t}} = X_{\bar{t}}^* \subseteq R^{\bar{t}}$ in case of the scarcity-situation, respectively, for $\bar{t} = 1, 2, \cdots, t$. Note that it is possible that $S^1 \notin \hat{A}_k(S^0, R^1)$.

On the other hand, it is simple to show that a sequence of configurations

$$U^0 = S^0, \quad U^1, \quad \cdots, \quad U^{t-1}, \quad U^t, \quad (S =: U^{t+1})$$

and a sequence of multisets

$$Y_1^*, \quad Y_2^*, \quad \cdots, \quad Y_{t-1}^*, \quad Y_t^*,$$

exist so that $U^{\bar{t}} = U^{\bar{t}-1} - Y_{\bar{t}}^* + R^{\bar{t}} \supseteq R^{\bar{t}}$ and $|Y_{\bar{t}}^*| = |R^{\bar{t}}|$ in case of the surplus-situation or $U^{\bar{t}} = Y_{\bar{t}}^* \subseteq R^{\bar{t}}$ in case of the scarcity-situation, respectively, for $\bar{t} = 1, 2, \cdots, t$ and where $w_t(S) = w_t(U^t) + D(U^t, S)$. [3]

Clearly, that $U^{\bar{t}}$ and $S^{\bar{t}}, \bar{t} \in \{1, 2, \cdots, t\}$ can be different and that

$$w_t(S) = \sum_{\bar{t}=0}^{t} D(S^t, S^{t+1}) = \sum_{\bar{t}=0}^{t} D(U^t, U^{t+1}).$$

The generalized work functions are quasi-convex. We say that a work function $w(= w_{\sigma_t})$ is quasi-convex if for any configurations X and Y and any subset $\tilde{X} \subseteq X$,

$$\min_{\tilde{Y} \subseteq Y, |\tilde{Y}| = |\tilde{X}|} \{w(X - \tilde{X} + \tilde{Y}) + w(Y - \tilde{Y} + \tilde{X})\} \leq w(X) + w(Y). \tag{15}$$

Lemma 1. *(Quasi-convexity lemma)*

All generalized work functions are quasi-convex.

This Lemma is shown in analogue to the proof of Lemma 10.4 in A. Borodin, R. El-Yaniv (1998). The case of the scarcity-situation must be considered in addition. The complete proof can be found in Appendix B.

---

[3] Pay attention to the fact that working with lazy algorithms is sufficient.

4 Considerations concerning the natural generalization of the work function algorithm

The work function algorithm (WFA) applied to the usual k-server problem can be found in A. Borodin, R. El-Yaniv (1998), p. 166 for example. A natural generalization of this algorithm, adapted to the k-server problem with parallel requests, is the following:

Let σ_t be the request sequence thus far and let S'^t be the configuration of the WFA algorithm after servicing σ_t. Then, given the next request R^{t+1}.

In the case of the surplus-situation algorithm WFA serves R^{t+1} with servers $X^{*W} \subseteq S'^t$ satisfying

$$X^{*W} = \arg \min_{X \subseteq S'^t, |X| = |R^{t+1}|} \{w_t(S'^t - X + R^{t+1}) + D(R^{t+1}, X)\}. \\ S'^{t+1} = S'^t - X^{*W} + R^{t+1} \tag{16}$$

is then the configuration at the end of step $t + 1$. According to (7), we can also use the equivalent formulas

$$\hat{X}^{*W} = \arg \min_{\hat{X} \subseteq \hat{S}'^t, \ |\hat{X}| = |\hat{R}^{t+1}|} \{w_t(S'^t - \hat{X} + \hat{R}^{t+1}) + D(\hat{R}^{t+1}, \hat{X})\} \tag{16a} \\ S'^{t+1} = S'^t - \hat{X}^{*W} + \hat{R}^{t+1}.$$

In the case of the scarcity-situation algorithm WFA chooses servers $X^{*W} \subseteq S'^t$ satisfying

$$X^{*W} = \arg \min_{X \subseteq R^{t+1}, |X| = k} \{w_t(X) + D(X, S'^t)\}. \\ \text{Then } S'^{t+1} = X^{*W}. \tag{17}$$

(An equivalent representation according to (11) would also be possible.)

Clearly, the cost of algorithm WFA to serve the request R^{t+1} is $D(R^{t+1}, X)$ ($= D(\hat{R}^{t+1}, \hat{X})$) in both cases.

Next we want to give an example that the WFA algorithm is not competitive in general (where the case $|R| > k$ is allowed).

Example 1 *Let $\epsilon \in (0, 1)$ and let $h = 1, 3, \cdots$ be a sequence of positive integers with $h_1 = 1$ and $h_{t+1} = h_t + t + 1$ for $t = 1, 2, \cdots$. We focus on a number of similar examples $E_2, E_3, \cdots, E_j, \cdots$ with $k = 1$, $S^0 = \{0\}$ and request sequences $\sigma^{n_j} = R^1, R^2, \cdots, R^{n_j}$, $n_j \in \{2, 3, \cdots, j\}$ with $R^t = \{-t - \epsilon, h_t\}$ for $t = 1, 2, \cdots$. Finally, the usual metric of the reals is used in all examples.*

*Then, $w_\emptyset(\{-1 - \epsilon\}) = 1 + \epsilon, w_\emptyset(\{h_1\}) = 1$,
$w_t(\{-t - \epsilon\}) = t + \epsilon, w_t(\{h_t\}) = h_t$ for $t = 1, 2, \cdots, n_j, j = 2, 3, \cdots$,
$w_t(\{-1 - t - \epsilon\}) = 1 + t + \epsilon, w_t(\{h_{t+1}\}) = h_{t+1}$ for $t = 1, 2, \cdots, n_j - 1$,
$j = 2, 3, \cdots$.*

The minimum costs of the corresponding offline problems are
$\text{cost(offline)} = \min_S w_\sigma(S) = w_{n_j}(\{-n_j - \epsilon\}) = n_j + \epsilon$ *for $j = 2, 3, \cdots$, where*
$S^{n_j} = \{-n_j - \epsilon\}$ *are the optimal solutions.*

Now, we show by mathematical induction that algorithm WFA yields $S'^t = \{h_t\}$ for $t = 1, 2, \cdots, n_j, j = 2, 3, \cdots$.

Base case. Obviously, $S'^1 = \{h_1\} = \{1\}$.

Induction step. Assume that $S'^t = \{h_t\}$.

According to (17),
$$X^{*W} = \arg \min_{X \subseteq \{-1-t-\epsilon, h_{t+1}\}, |X|=1} \{w_t(X) + D(X, \{h_t\})\}.$$
Since the definition of h implies
$$w_t(\{-1-t-\epsilon\}) + D(\{-1-t-\epsilon\}, \{h_t\}) = 1 + t + \epsilon + 1 + t + \epsilon + h_t$$
$$> w_t(\{h_{t+1}\}) + D(\{h_{t+1}\}, \{h_t\}) = h_{t+1} + h_{t+1} - h_t,$$
$$X^{*W} = S'^{t+1} = \{h_{t+1}\} \text{ follows.}$$

Then the cost of algorithm WFA to serve all requests is
$$\text{cost(WFA)} = 1 + \sum_{t=1}^{n_j-1} D(\{h_{t+1}\}, \{h_t\}) = \sum_{t=1}^{n_j} t = \frac{n_j \ (n_j+1)}{2} \text{ for any example } E_j.$$
From this we get the following equation
$$\frac{\text{cost(WFA)}}{\text{cost(offline)}} = \frac{1/2 \ n_j \ (n_j+1)}{n_j+\epsilon}. \text{ Algorithm WFA applied to the k-server problem with}$$
parallel request is not competitive (if the case $|R| > k$ is allowed), since
$$\lim_{n_j \to \infty} \frac{1/2 \ n_j \ (n_j+1)}{n_j+\epsilon} = \infty.$$

From now on, we will focus on the surplus-situation.

Remark 2 *The definitions of the minimizer, the maximizer and the potential-like function are used to prove that the WFA algorithm applied to the usual k-server problem is competitive (see A. Borodin, R. El-Yaniv (1998), Section 10.7). These definitions can be adapted to the k-server problem with parallel request and the natural generalization of the WFA. However, it is not possible to complete a proof for competitiveness of the natural generalization of the WFA (where only the surplus-situation is allowed) with success in a similar way as in A. Borodin, R. El-Yaniv (1998). The reason for that is certain partial results are not valid (in general), for example, an inequality which corresponds to (10.22) in A. Borodin, R. El-Yaniv (1998), p. 170.*

So far, the answer to the question, whether the natural generalization of the WFA is competitive or not in the case of the surplus-situation is a difficult open problem. Thats why, we will suggest the new "compound work function algorithm" in the following section and prove that this algorithm is also $(2k - 1)$-competitive.

5 The compound work function algorithm

As a basis for further considerations we introduce the following more specific k-server problem. The term "k-server problem with a simple extension" describes a k-server problem with the following properties:

(i) More than one server can be located on a point.

(ii) Requests R^t with $R^t(r) \neq \emptyset$ for one $r \in M$ and where in addition, $|R^t(r)| \leq \min\{|R^{t-1}(r)| + 1, k\}$ must be fulfilled, are allowed.

Then at most one server must be moved in servicing the request in each step.

Lemma 2. *Algorithm WFA applied to the k-server problem with a simple extension is $(2\,k-1)$-competitive for any k and any metric space.*

This lemma can be shown in analogue to the proof of Theorem 10.9 in A. Borodin, R. El-Yaniv (1998). Therefore, we do not show the complete proof, but make the following comments (see also Remark 2). We define the minimizer configuration A of p with respect to w_t similarly as in A. Borodin, R. El-Yaniv (1998):

$A = \arg\min_X\{w_t(X) - \sum_{i \in M} |X(i)|\ d(i,p)\}$, where w_t is the current work function and $p \in M$. The correctness of an inequality which corresponds to (10.22) in A. Borodin, R. El-Yaniv (1998), p. 170. can be proved, since $\quad D(\tilde{X}, R) = \sum_{i \in M} |\tilde{X}(i)|\ d(i,r)$, where R is a request with $R(r) \neq \emptyset, R(i) = \emptyset$ for $i \neq r$ and $\tilde{X}$ is a multiset with $|\tilde{X}| = |R| = |R(r)|$.

The compound work function algorithm is based on a surrogate problem, which is a k-server problem with a simple extension.

In order to construct a surrogate problem we replace the steps of the original problem (k-server problem with parallel request, where only the surplus-situation is allowed) by a number of steps in the surrogate problem. In more detail, let $R^t = \{p_1, \cdots, p_1, p_2, \cdots, p_2, ..., p_m, ..., p_m\}$ with $p_i \neq p_j$ for $i \neq j$ be the request in the t-th step.

Then we set

$$\bar{m} := |\{r \in M \mid R^t(r) \neq \emptyset\}| \tag{18}$$

and

$$\rho_i := |R^t(p_i)|, \bar{\rho}_i := \sum_{l=1}^{i} \rho_l \text{ for } i = 1, \cdots, m, \bar{\rho}_0 := 0, \bar{\rho} := \bar{\rho}_m. \tag{19}$$

(Note that $\bar{m}$ and $\bar{\rho}, \cdots$ depend on t.)

Furthermore, let $\bar{j}$ be an integer with $\bar{j} = 1$ if $\bar{m} = 1$ and with

$$\delta_* \cdot (\bar{j} - 1) > \bar{\rho} \cdot \delta^* = |R^t| \cdot \delta^*, \text{ if } \bar{m} > 1, \text{ respectively,}$$
$$\text{where } \delta_* = \min_{\{r_1, r_2\} \subseteq R^t, r_1 \neq r_2} d(r_1, r_2), \delta^* = \max_{r \in R^t, s \in \bar{M}} d(r,s) \tag{20}$$

$(\bar{M}$ as defined in (14)).

((20) is a sufficient condition for the validity of the following Lemma 3 and Lemma 4.)

We replace a step t of the generalized k-server problem by steps $t_{1,1}, \cdots, t_{1,\bar{\rho}}, t_{2,1}, \cdots, t_{2,\bar{\rho}}, \cdots, t_{\bar{j},1}, \cdots, t_{\bar{j},\bar{\rho}}$ with requests $\bar{R}^{1,1}, \cdots, \bar{R}^{1,\bar{\rho}}, \bar{R}^{2,1}, \cdots, \bar{R}^{2,\bar{\rho}}, \cdots, \bar{R}^{\bar{j},1}, \cdots, \bar{R}^{\bar{j},\bar{\rho}}$ in the surrogate problem, where

$$|\bar{R}^{j,f}(p_i)| = \begin{cases} f - \bar{\rho}_{i-1} & \text{for } i \text{ with } R^t(p_i) \neq \emptyset, f \in \{\bar{\rho}_{i-1} + 1, \bar{\rho}_{i-1} + 2, \cdots, \bar{\rho}_i\}, \\ & \text{and } j = 1, \cdots, \bar{j} \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

Since $\bar{R}^{j,f}$ are independent of j we set

$$\bar{R}^f := \bar{R}^{j,f} \text{ for } f = 1, 2, \cdots, \bar{\rho}. \tag{22}$$

For example, $R^t = \{p_1, p_1, p_2, p_2, p_2\}$ implies that $\bar{m} = 2, \bar{\rho} = 5$ and $\bar{R}^1 = \{p_1\}$, $\bar{R}^2 = \{p_1, p_1\}, \bar{R}^3 = \{p_2\}, \bar{R}^4 = \{p_2, p_2\}, \bar{R}^5 = \{p_2, p_2, p_2\}$.

Work functions for the surrogate problem are indexed in correspondence to the steps and the requirements: $w_{t_{j,f}}$. In addition, we set

$$w_t := w_{t_{\bar{j},\bar{\rho}}} \text{ for } t = 1, 2, \cdots, m \text{ and } w_{t_{\bar{j},0}} := w_{t_{\bar{j}-1,\bar{\rho}}} \text{ if } \bar{j} > 1. \tag{23}$$

In the following we characterize the solutions of the surrogate problem considered first as offline problem, where the entire sequence of requests is known in advance, and second as online problem.

Lemma 3. *(offline). Let (SP) be the surrogate problem for a given k-server problem with parallel requests. Furthermore, let S be any configuration and $\bar{z} \in \{1, \cdots, \bar{\rho}\}$. If, according to (6), $w_{t_{\bar{j},\bar{z}}}(S) = w_{t_{\bar{j},\bar{z}-1}}(S - Y^*_{\bar{j},\bar{z}} + \bar{R}^{\bar{z}}) + D(Y^*_{\bar{j},\bar{z}}, \bar{R}^{\bar{z}})$ for a $Y^*_{\bar{j},\bar{z}} \subseteq S$ with $\left| Y^*_{\bar{j},\bar{z}} \right| = |\bar{R}^{\bar{z}}|$ then $Y^*_{\bar{j},\bar{z}}$ satisfies the following property*

$$Y^*_{\bar{j},\bar{z}} \cap R^t = S \cap \bar{R}^{\bar{z}}. \tag{24}$$

PROOF. Clearly, if $\bar{m} = 1$ then (24) is fulfilled. Now, we want to consider the case that $\bar{m} > 1$: Let $R^t(i_l) \neq \emptyset$ for $l = 1, 2, \cdots, \bar{m}$ and $\bar{l} \in \{1, \cdots, \bar{m}\}$ such that $\bar{\rho}_{i_{\bar{l}-1}} < \bar{z} \leq \bar{\rho}_{i_{\bar{l}}}$. Further on, let

$$\begin{aligned}
w_{t_{\bar{j},\bar{z}}}(S) = w_{t-1}(&S - Y^*_{\bar{j},\bar{z}} + \bar{R}^{\bar{z}} - Y^*_{\bar{j},\bar{z}-1} + \bar{R}^{\bar{z}-1} - + \cdots - Y^*_{\bar{j},1} + \bar{R}^1 \\
&- Y^*_{\bar{j}-1,\bar{\rho}} + \bar{R}^{\bar{\rho}} - Y^*_{\bar{j}-1,\bar{\rho}-1} + \bar{R}^{\bar{\rho}-1} - + \cdots - Y^*_{\bar{j}-1,1} + \bar{R}^1 - + \cdots \\
&- Y^*_{1,\bar{\rho}} + \bar{R}^{\bar{\rho}} - Y^*_{1,\bar{\rho}-1} + \bar{R}^{\bar{\rho}-1} - + \cdots - Y^*_{1,1} + \bar{R}^1) \\
&+ D(Y^*_{1,1}, \bar{R}^1) + \cdots + D(Y^*_{1,\bar{\rho}-1}, \bar{R}^{\bar{\rho}-1}) + D(Y^*_{1,\bar{\rho}}, \bar{R}^{\bar{\rho}}) + \cdots \\
&+ D(Y^*_{\bar{j}-1,1}, \bar{R}^1) + \cdots + D(Y^*_{\bar{j}-1,\bar{\rho}-1}, \bar{R}^{\bar{\rho}-1}) + D(Y^*_{\bar{j}-1,\bar{\rho}}, \bar{R}^{\bar{\rho}}) \\
&+ D(Y^*_{\bar{j},1}, \bar{R}^1) + \cdots + D(Y^*_{\bar{j},\bar{z}-1}, \bar{R}^{\bar{z}-1}) + D(Y^*_{\bar{j},\bar{z}}, \bar{R}^{\bar{z}})
\end{aligned}$$

be a recursive representation, where $Y^*_{\bar{j},\bar{z}}, \cdots, Y^*_{1,1}$ are computed according to (6). The minimum weight matchings between $Y^*_{\bar{j},\bar{z}}$ and $\bar{R}^{\bar{z}}, \cdots, Y^*_{1,1}$ and $\bar{R}^1$ yield the optimal movements of the servers in the steps $t_{1,1}, \cdots, t_{\bar{j},\bar{z}}$. Let ς be the number of real movements (with distances greater than 0) from $S - R^t$ in R^t which occur in these steps. Then the definition of the work functions together with the triangle-inequality imply that $\varsigma \leq |R^t - S|$ and that these movements must be executed in the first $\bar{\rho}$ steps backwards. If $\varsigma = |R^t - S|$ then the total cost is not greater than $|R^t - S| \cdot \delta^*$.

If $\varsigma < |R^t - S|$ then at least one real movement of a server from R^t in R^t in all subsequences of steps $(t_{j,\bar{z}}, \cdots, t_{j-1,\bar{z}+1}), 2 \leq j \leq \bar{j}$ is necessary. The total cost would in this case be greater or equal to $\delta_* \cdot (\bar{j} - 1)$. Thus, from condition (20) it follows that $\varsigma = |R^t - S|$.

The above properties of optimal movements imply that

$$Y^*_{\bar{j},z} \cap R^t = S \cap \bar{R}^z \text{ for } z \in \{\bar{\rho}_{i_1}, \cdots, \bar{\rho}_{i_{\bar{l}-1}}, \bar{z}\},$$

$$Y^*_{\bar{j},z} = \bar{R}^z \text{ for } z \in \{1, \cdots, \bar{z}-1\} \setminus \{\bar{\rho}_{i_1}, \cdots, \bar{\rho}_{i_{\bar{l}-1}}\},$$

$$Y^*_{\bar{j}-1,z} \cap R^t = S \cap \bar{R}^z \text{ for } z \in \{\bar{\rho}_{i_{\bar{l}+1}}, \cdots, \bar{\rho}_{i_{\bar{m}}}(=\bar{\rho})\},$$

$$Y^*_{\bar{j}-1,z} \cap R^t = (S + (\bar{R}^{\bar{z}} - S)) \cap \bar{R}^z \text{ for } z = \bar{\rho}_{i_{\bar{l}}},$$

$$Y^*_{\bar{j}-1,z} = \bar{R}^z \text{ for } z \in \{1, \cdots, \bar{\rho}\} \setminus \{\bar{\rho}_{i_{\bar{l}+1}}, \cdots, \bar{\rho}_{i_{\bar{m}}}\} \text{ and}$$

$$Y^*_{j,z} = \bar{R}^z \text{ for } j \in \{1, \cdots, \bar{j}-2\}, z \in \{1, \cdots, \bar{\rho}\}.$$

Thus, the statement $Y^*_{\bar{j},z} \cap R^t = S \cap \bar{R}^z$ for $z = \bar{z}$ is shown. $\qquad\square$

Lemma 4. *(offline). Let (SP) be the surrogate problem for a given k-server problem with parallel requests (OP) and let the entire sequence of requests be known in advance.*

(i) If S is the configuration of the offline servers at the beginning of a step $t_{1,1}$ corresponding to an optimal solution of (SP), then such a optimal solution of (SP)
   (∗) includes exactly $|R^t - S|$ real movements in the steps $t_{1,1}, \cdots, t_{\bar{j},\bar{\rho}}$. All these movements are movements from $S - R^t$ in R^t, and are executed in the first $\bar{\rho}$ steps $t_{1,1}, \cdots, t_{1,\bar{\rho}}$.
(ii) The optimal offline costs of (SP) and (OP) are equal.

    PROOF.

    (i) can be proved in a similar way as Lemma 3 (see also Remark 1).

    (ii) follows from (i) and the facts that a feasible solution of (OP) implies a feasible solution of (SP) which fulfilled properties (*) and vice versa. $\qquad\square$

Theorem 1. *(online). Let (SP) be the surrogate problem for a given k-server problem with parallel requests. Then a sequence of online servers' configurations exists which can be constructed by the WFA algorithm so that $R^t \subseteq S'^{\bar{j},\bar{\rho}}$ for any t, where $S'^{\bar{j},\bar{\rho}}$ denotes the online servers configuration at the end of step $t_{\bar{j},\bar{\rho}}$.*

    PROOF. If $\bar{m} = 1$ then the proof is obvious. Now, we want to consider the case $\bar{m} > 1$: Let $R^t(i_l) \neq \emptyset$ for $l = 1, 2, \cdots, \bar{m}$. In addition, we set $\bar{\rho}_{i_0} := 0$ and $\bar{R}^0 := \emptyset$. $R^t \subseteq S'^{\bar{j},\bar{\rho}}$ is equivalent to $\bar{R}^{\bar{\rho}_{i_1}} + \bar{R}^{\bar{\rho}_{i_2}} + \cdots + \bar{R}^{\bar{\rho}_{i_{\bar{m}}}} \subseteq S'^{\bar{j},\bar{\rho}}$.

For this we prove by mathematical induction that
$$\bar{R}^{\bar{\rho}_{i_0}} + \bar{R}^{\bar{\rho}_{i_1}} + \cdots + \bar{R}^{\bar{\rho}_{i_{\bar{l}-1}}} + \bar{R}^{\bar{z}} \subseteq S'^{\bar{j},\bar{z}} \text{ for each } \bar{z} \in \{1, \cdots, \bar{\rho}\}, \text{ where}$$
$$\bar{l}(= \bar{l}(\bar{z})) \in \{1, \cdots, \bar{m}\} \text{ such that } \bar{\rho}_{i_{\bar{l}-1}} < \bar{z} \leq \bar{\rho}_{i_{\bar{l}}}.$$

Base case. Obviously, $\bar{R}^1(= \{i_1\}) \subseteq S'^{\bar{j},1}$.

Induction step. Assume that $\bar{R}^{\bar{\rho}_{i_1}} + \bar{R}^{\bar{\rho}_{i_2}} + \cdots \cup \bar{R}^{\bar{\rho}_{i_{\bar{l}-1}}} + \bar{R}^f \subseteq S'^{\bar{j},f}$ for
$$f \in \{\bar{\rho}_{i_{\bar{l}-1}} + 1, \bar{\rho}_{i_{\bar{l}-1}} + 2, \cdots, \bar{\rho}_{i_{\bar{l}}}\} \text{ if } \bar{l} \in \{1, \cdots, \bar{m}-1\} \text{ or}$$
$$f \in \{\bar{\rho}_{i_{\bar{m}-1}} + 1, \bar{\rho}_{i_{\bar{m}-1}} + 2, \cdots, \bar{\rho}_{i_{\bar{m}}} - 1\}, \text{ respectively. There } S'^{\bar{j},f} \text{ denotes the}$$
online servers' positions at the end of step $t_{\bar{j},f}$. In addition, we notice that $|\bar{R}^f(i_{\bar{l}})| = f - \bar{\rho}_{i_{\bar{l}-1}}$ according to (21).

At first, we consider the case that $f < \bar{\rho}_{i_{\bar{l}}}$.

Then $|\bar{R}^{f+1}(i_{\bar{l}})| = f + 1 - \bar{\rho}_{i_{\bar{l}-1}} = |\bar{R}^f(i_{\bar{l}})| + 1$ is valid.

We will show that

$$\bar{R}^{\bar{\rho}_{i_0}} + \bar{R}^{\bar{\rho}_{i_1}} + \cdots + \bar{R}^{\bar{\rho}_{i_{\bar{l}-1}}} + \bar{R}^{f+1} \subseteq S'^{\bar{j},f+1}. \tag{25}$$

If already $\bar{R}^{f+1} \subseteq S'^{\bar{j},f}$ then $S'^{\bar{j},f} = S'^{\bar{j},f+1}$ follows and (25) is true. Now, let $\bar{R}^{f+1} \nsubseteq S'^{\bar{j},f}$.

Then a $S'^{\bar{j},f+1} = S'^{\bar{j},f} - \hat{X}^{*W} + i_{\bar{l}}$, which satisfies (25), can be determined by the WFA algorithm, where $\hat{X}^{*W} \subseteq \hat{S}'^{\bar{j},f}$ with $\hat{X}^{*W} \neq \{i_{\bar{l}}\}$ and $\left|\hat{X}^{*W}\right| = 1$ is computed as an element from the argminset of:

$$w_{\bar{j},f}(S'^{\bar{j},f} - \hat{X} + i_{\bar{l}}) + D(\hat{X}, \{i_{\bar{l}}\}) \tag{26}$$

over $\hat{X}$, according to (16a).

Case I: $\left\{\hat{X}^{*W} \cap \bar{R}^{\bar{\rho}_{i_l}} = \emptyset \text{ for all } l \in \{1, 2, \cdots \bar{l} - 1\}\right\}$ or

$\left\{ \text{if } \hat{X}^{*W} = \{i_l\} \, (l \in \{1, 2, \cdots \bar{l} - 1\}) \text{ then also } \bar{R}^{\bar{\rho}_{i_l}} + i_l \subseteq S'^{\bar{j},f}\right\}$:

In this case (25) follows immediately.

Case II: $\hat{X}^{*W} = \{i_\lambda\} \, (\lambda \in \{1, 2, \cdots \bar{l} - 1\})$ and $\bar{R}^\varphi + i_\lambda \nsubseteq S'^{\bar{j},f}$

for $\varphi = \bar{\rho}_{i_\lambda}$:

In this case we will show that then also a $\hat{X}^{*1}(\neq i_{\bar{l}})$ exists which satisfies Case I and implies the same value of (26) as $\hat{X}^{*W}$.

(26) leads to

$$w_{t_{\bar{j},f}}(S'^{\bar{j},f} - i_\lambda + i_{\bar{l}}) + D(\{i_\lambda\}, \{i_{\bar{l}}\}) \tag{27}$$

for $\hat{X}^{*W} = \{i_\lambda\}$. In addition,

$$w_{t_{\bar{j},f}}(S'^{\bar{j},f} - i_\lambda + i_{\bar{l}}) = w_{t_{\bar{j},\varphi}}(S'^{\bar{j},f} - i_\lambda + i_{\bar{l}}), \tag{28}$$

since $\bar{R}^u \subseteq S'^{\bar{j},f} - i_\lambda + i_{\bar{l}}$ for $u = \varphi + 1, \varphi + 2, \cdots, f$ according to the induction hypothesis and Case II.

Using formula (6) for the recursive computation of work functions yields

$$\begin{aligned} w_{t_{\bar{j},\varphi}}(S'^{\bar{j},f} - i_\lambda + i_{\bar{l}}) \\ = \min_{Y \subseteq S'^{\bar{j},f} - i_\lambda + i_l, |Y| = \rho_{i_\lambda}} \{ w_{t_{\bar{j},\varphi-1}}(S'^{\bar{j},f} - i_\lambda + i_{\bar{l}} - Y + \{\rho_{i_\lambda}\}) \\ + D(Y, \{\rho_{i_\lambda}\}) \}, \end{aligned} \tag{29}$$

where $\{\rho_{i_\lambda}\}$ denotes the multiset which only include a number of ρ_{i_λ} elements i_λ. Lemma 3 (with $S = S'^{\bar{j},f} - i_\lambda + i_{\bar{l}}$ and $\bar{z} = \varphi$) implies that

$$Y^* \cap R^t = (S'^{\bar{j},f} - i_\lambda + i_{\bar{l}}) \cap \bar{R}^\varphi \tag{30}$$

for optimal solutions Y^* of (29). In Case II this means

$$Y^* \cap R^t = \{\rho_{i_\lambda} - 1\} \text{ and hence,}$$

$$Y^* = \{\rho_{i_\lambda} - 1\} + \hat{Y}^* \text{ with } |\hat{Y}^*| = 1 \text{ and}$$

$$\hat{Y}^* \neq \{i_{\bar{\lambda}}\} \ (\bar{\lambda} \in \{1, 2, \cdots, \bar{l} - 1\}) \text{ if } \bar{R}^{\bar{\varphi}} + i_{\bar{\lambda}} \nsubseteq S'^{\bar{j}, f} \text{ for } \bar{\varphi} = \bar{\rho}_{i_{\bar{\lambda}}}. \quad (31)$$

Furthermore,

$$\hat{Y}^* \neq \{i_{\bar{l}}\} \quad (32)$$

also follows from (30). Hence (29) can be written as
$$w_{t_{\bar{j}, \varphi}}(S'^{\bar{j}, f} - i_\lambda + i_{\bar{l}}) = w_{t_{\bar{j}, \varphi - 1}}(S'^{\bar{j}, f} + i_{\bar{l}} - \hat{Y}^*) + D(\hat{Y}^*, \{i_\lambda\}).$$
This equation (see also (28)) is equivalent to

$$w_{t_{\bar{j}, f}}(S'^{\bar{j}, f} - i_\lambda + i_{\bar{l}}) = w_{t_{\bar{j}, f}}(S'^{\bar{j}, f} + i_{\bar{l}} - \hat{Y}^*) + D(\hat{Y}^*, \{i_\lambda\}), \quad (33)$$

since $\bar{R}^u \subseteq S'^{\bar{j}, f} - i_\lambda + i_{\bar{l}}$ for $u = \varphi + 1, \varphi + 2, \cdots, f$ according to the induction hypothesis and (31) and (32).

Now, we want to determine the argminset of (26) where in addition only $\hat{X}$ with $\hat{X} \neq \{i_{\bar{\lambda}}\}$ $(\bar{\lambda} \in \{1, 2, \cdots \bar{l} - 1\})$ if $\bar{R}^{\bar{\rho}_{i_{\bar{\lambda}}}} + i_{\bar{\lambda}} \nsubseteq S'^{\bar{j}, f}$ (see Case I) are allowed. Let $\hat{X}^{*1}$ denote an element of such an argminset.

These conditions are also fulfilled by $\hat{Y}^*$ according to (31), (32) and $\bar{R}^{\bar{\rho}_{i_{\bar{\lambda}}}} + i_\lambda \nsubseteq S'^{\bar{j}, f}$ in Case II. Hence,

$$w_{\bar{j}, f}(S'^{\bar{j}, f} - \hat{X}^{*1} + i_{\bar{l}}) + D(\hat{X}^{*1}, \{i_{\bar{l}}\}) \leq w_{\bar{j}, f}(S'^{\bar{j}, f} - \hat{Y}^* + i_{\bar{l}}) + D(\hat{Y}^*, \{i_{\bar{l}}\}). \quad (34)$$

(34) and (33) together with the triangle-inequality
$-D(\hat{Y}^*, \{i_\lambda\}) + D(\hat{Y}^*, \{i_{\bar{l}}\}) \leq D(\{i_\lambda\}), \{i_{\bar{l}}\})$, imply that

$$w_{t_{\bar{j}, f}}(S'^{\bar{j}, f} + i_{\bar{l}} - \hat{X}^{*1}) + D(\hat{X}^{*1}, \{i_{\bar{l}}\}) \leq w_{t_{\bar{j}, f}}(S'^{\bar{j}, f} + i_{\bar{l}} - i_\lambda) + D(\{i_\lambda\}), \{i_{\bar{l}}\}).$$

$\hat{X}^{*W} = \{i_\lambda\}$ is an element of the argminset in Case II. However, the last inequality shows that in every case an element $\hat{X}^{*1}$ of the argminset exists which satisfies Case I.

In the case that $f = \bar{\rho}_{i_{\bar{l}}}$ $(\bar{l} \in \{1, \cdots, \bar{N} - 1\})$ it is $|\bar{R}^{f+1}(i_{\bar{l}+1})| = 1$. Then $\bar{R}^{\bar{\rho}_{i_1}} + \bar{R}^{\bar{\rho}_{i_2}} + \cdots + \bar{R}^{\bar{\rho}_{i_{\bar{l}}}} + \bar{R}^{f+1} \subseteq S'^{\bar{j}, f+1}$ can be shown in analogue to the above proof (in the case that $f < \bar{\rho}_{i_{\bar{l}}}$), where all $i_{\bar{l}}$ are to be replaced by $i_{\bar{l}+1}$ and all sets of indices $\{1, 2, \cdots \bar{l} - 1\}$ by $\{1, 2, \cdots \bar{l}\}$. $\qquad \square$

Now we want to suggest **the compound work function algorithm** (compound WFA):

(0) Let (OP) be a k-server problem with parallel requests, where only the surplus-situation is allowed.
(1) Construct a corresponding surrogate problem (SP) according to (18), ..., (23) (which is a k-server problem with a simple extension).
(2) Solve (SP) by means of the usual WFA.
(3) Skip the movements of servers, which are unnecessary for the original problem (OP).

In more detail, let σ_{t-1} be the request sequence thus far and let S'^{t-1} be the configuration of WFA after servicing σ_{t-1}. Then, given the next request R^t.

Compound WFA, (step t): *Apply the (usual) WFA algorithm to the surrogate problem in the steps $t_{1,1}, \cdots, t_{1,\bar\rho}, t_{2,1}, \cdots, t_{2,\bar\rho}, t_{\bar{j},1}, \cdots, t_{\bar{j},\bar\rho}$, which replace step t of the original problem, and construct a sequence of online servers' configurations*
$$S'^{1,1}, S'^{1,2}, \cdots, S'^{1,\bar\rho}, S'^{2,1}, S'^{2,2}, \cdots, S'^{2,\bar\rho}, \cdots, S'^{\bar{j},1}, S'^{\bar{j},2}, \cdots, S'^{\bar{j},\bar\rho}$$
such that $R^t \subseteq S'^{\bar{j},\bar\rho}$. Set $S'^t := S'^{\bar{j},\bar\rho}$. [4]

Skip movements of servers, which are unnecessary for the original problem, in the following way: The set of all (real) movements of the servers in the surrogate steps can be partitioned that if $|S'^t(\bar{n})| > |S'^{t-1}(\bar{n})|$ then movements of $|S'^t(\bar{n})| - |S'^{t-1}(\bar{n})|$ servers exist for each of them as follows. Such a server is located on a point n_0 with $|S'^t(n_0)| < |S'^{t-1}(n_0)|$ at the beginning of step t. In the steps: $t_{j_{n_0}, f_{n_0}}, t_{j_{n_1}, f_{n_1}}, \cdots,$ $t_{j_{n_{\bar\imath}}, f_{n_{\bar\imath}}}$ (where $j_{n_0} \le j_{n_1} \le \cdots \le j_{n_{\bar\imath}}, n_{\bar\imath} = \bar{n}$ and $\bar\rho_{n_i - 1} < f_{n_i} \le \bar\rho_{n_i}$) the server is moved from point n_0 to point n_1 and so on, finally to point $n_{\bar\imath}$. Replace these movements by one. Move the corresponding server directly from point n_0 to point $\bar{n}$.

We want to point out that the condition $R^t \subseteq S'^t$ for any t can be fulfilled according to Theorem 1. This condition implies that the compound WFA algorithm solves the original problem (OP) (the k-server problem with parallel requests).

Theorem 2. *The compound WFA algorithm is $(2\,k-1)$-competitive for any k and any metric space.*

PROOF. According to Lemma 2, WFA is $(2\,k-1)$-competitive for the k-server problem with a simple extension, thus also for (SP). If the number of movements is reduced by the compound WFA then the online cost is lower because of the triangle-inequality. The optimal offline costs of (SP) and (OP) are equal (Lemma 4, (ii)). Hence, the compound WFA algorithm is also $(2\,k-1)$-competitive. □

Finally, let us remark that more values of work functions are used by compound WFA than by WFA from Section 4. Values of work functions are important for the competitiveness of corresponding algorithms. In Appendix C, at least an example can be found, where compound WFA yield a better result than WFA.

6 Conclusion

In this paper we have considered a generalized k-server problem with parallel requests. We have defined corresponding work functions. The work functions are quasi-convex as well in the case of the surplus-situation as in the case of the scarcity-situation. A corresponding algorithm WFA is not competitive in the case that the scarcity-situation is allowed. The question, whether this algorithm is competitive or not in the case of the surplus-situation is a difficult open problem. We have constructed the compound WFA algorithm which uses more values of work functions than algorithm WFA from Section 4. We were able to prove the same bound of the competitive ratio as for the WFA algorithm applied to the (usual) k-server problem.

---

[4] Clearly, if $R^t \subseteq S'^{j,\rho}$ for $j < \bar{j}$ or $\rho < \bar\rho$ then $S'^{\bar{j},\bar\rho} = S'^{\bar{j},\bar\rho-1} = \cdots = S'^{j,\rho+1} = S'^{j,\rho}$. Hence, $S'^t := S'^{j,\rho}$ is already the end configuration of step t.

References

Bartal, Y., Grove, E. (2000). The Harmonic k-Server Algorithm Is Competive. *Journal of the ACM* 47(1), 1 - 15.

Borodin, A., El-Yaniv, R. (1998). *Online computation and competitive analysis*. University Press, Cambrigde.

Borodin, A., Linial,N., Saks, M. (1992). An optimal online algorithm for metrical task systems. *Journal of the ACM* 39(4), 745-763.

Hildenbrandt, R. (1993). A special stochastic decision problem, *Optimization* 28, 95-110.

Hildenbrandt, R. (1995; Libri BoD 2000). *Methoden aus ganzzahliger Optimierung und Verbandstheorie zur Behandlung eines stochastischen dynamischen Transportproblems*. Habilitationsschrift, TU Ilmenau.

Hildenbrandt, R. (2014). A k-server problem with parallel requests and unit distances. *Information Processing Letters* 114(5), 239-246.
    doi: 10.1016/j.ipl.2013.12.011

Hildenbrandt, R. (2016). The k-Server Problem with Parallel Requests and the Compound Harmonic Algorithm. *Baltic J. Modern Comp.* 4(3), 607-629.

Koutsoupias, E. (1994). *On-line Algorithms and the k-Server Conjecture.* Ph.D.dissertation, Department of Computer Science and Engineering, University of California, San Diego.

Koutsoupias, E., Papadimitriou, C. (1995). On the k-server conjecture. *Journal of the ACM* 42(5), 971-983.

Krumke, S. O., Rambau, J. (2005). *Online Optimierung.* Vorlesungsskript, Technische Universität Berlin.
    www.fsmpi.uni-bayreuth.de/skripten/online_optimierung.

Manasse, M.S., McGeoch L.A., Sleator, D.D. (1988). Competitive algorithms for on-line problems. In: *Proceeding of the 20th Annual ACM Symposium on Theory of Computing*, 322-333.

Manasse, M.S., McGeoch L.A., Sleator, D.D. (1990). Competitive algorithms for server problems. *J. Algorithms* 11(2), 208230.

Sitters, R. (2013). The generalized work function algorithm is competitive for the generalized 2-server problem. *SIAM J. Comput.* 43(1), 96125.

Sleator, D. D., Tarjan, R. E. (1985). Amortized efficiency of list update and paging rules. *Communications of the ACM* 28(2), 202-208.

Appendices

A Computation of the set distance

If C is a multiset then let $C_M \subseteq M$ denote a set (no multiset) where $p \in C_M$ if and only if $p \in C$ at least once. The value $D(X,Y)$ of the minimum weight matching between X and Y (where $|X| = |Y|$) can then be computed by:

$$D(X,Y) = \min_{z_{p,r}, p \in X_M, r \in Y_M} \sum_{p \in X_M} \sum_{r \in Y_M} z_{p,r} d(p,r)$$
$$\text{subject to } \sum_{r \in Y_M} z_{p,r} = |X(p)| \; \forall \, p \in X_M,$$
$$\sum_{p \in X_M} z_{p,r} = |Y(r)| \; \forall \, r \in Y_M,$$

$$z_{p,r} \in \mathbb{Z}_+ \ \forall \ p \in X_M, r \in Y_M.$$

B Proof of the quasi-convexity lemma

Proof. In order to prove the lemma, we show that work functions satisfy a more general quasi-convexity property.

Property GQ: Let w be a work function and let X and Y be two configurations. Then there exists a bijection $g : X \to Y$ such that for all partitions of X into $\tilde{X}_1$ and $\tilde{X}_2$, the following holds

$$w(\tilde{X}_1 + g(\tilde{X}_2)) + w(g(\tilde{X}_1) + \tilde{X}_2) \le w(X) + w(Y). \tag{A.1}$$

By setting $\tilde{X}_1 = X - \tilde{X}(= X - \tilde{X}_2)$ and letting $\tilde{Y} = g(\tilde{X})(= g(\tilde{X}_2))$, we can see that the property GQ implies the quasi-convexity property. We prove that the work function satisfy the property GQ by relying the following claim.

Claim: If the bijection g satisfies equation (A.1), then there exists a bijection $\bar{g}$ such that $\bar{g}(x) = x$ for all $x \in X \cap Y$, which also satisfies equation (A.1).

Firstly, the preceding claim is proved. Let $g : X \to Y$ be a bijection. Assume that among all such bijections, g maps the maximum number of elements from $X \cap Y$ to themselves. Assume, by contradiction, that there exists some $a \in X \cap Y$ such that $g(a) \ne a$.

Define the bijection $\bar{g} : X \to Y$ that agrees with g everywhere but interchanges the values of g on a and $a^{-1} = g^{-1}(a)$.
That is, set $\bar{g}(a) = a$ and $\bar{g}(a^{-1}) = g(a)$.

By our assumption, $\bar{g}$ cannot satisfies property GQ.

Let (X_1, X_2) be any partition of X and assume, without loss of generality, that $a^{-1} = g^{-1}(a) \in X_1$.

It cannot be the case that, $a \in X_1$; if it were the case, then $g(X_1) = \bar{g}(X_1)$ and $g(X_2) = \bar{g}(X_2)$ and the inequality (A.1) would be fulfilled.

Given $a \notin X_1$ (hence, $a \in X_2$), we have

$$w(X) + w(Y) \ge w((X_1 + a) + g(X_2 - a)) + w((g(X_1 + a) + (X_2 - a)),$$
$$\text{since } g \text{ satisfies equation (A.1)};$$
$$= w((X_1 + a) + \bar{g}(X_2 - a)) + w((\bar{g}(X_1 + a) + (X_2 - a)),$$
$$\text{by the definition of } \bar{g};$$
$$= w(X_1 + \bar{g}(X_2)) + w(\bar{g}(X_1) + X_2).$$

Hence, $\bar{g}$ does satisfy property GQ, which is a contradiction.

Now, we prove the following by induction on the length of the request sequence.

Induction hypothesis: Work functions satisfy the property GQ.

Base case. For $t = 0$ we have $w_\emptyset(X) + w_\emptyset(Y) = D(S^0, X) + D(S^0, Y)$.
Consider two minimum weight matchings M_X and M_Y whose values are $D(S^0, X)$ and $D(S^0, Y)$, respectively. Each point $s_j \in S^0$ is mapped by M_X to some point

$x_j \in X$ and by M_Y to some point $y_j \in Y$. It is easy to see that the bijection $g(x_j) = y_j$ satisfies equation (A.1).

Induction step. Assume that w satisfies the property GQ and let R be the new request. We show that $w'(= w_{\sigma_t, R^{t+1}})$ satisfies the property GQ.

In the case of the surplus-situation there exists some $\tilde{X} \subseteq X$ with $\left| \tilde{X} \right| = |R|$, so that $w'(X) = w(X - \tilde{X} + R) + D(\tilde{X}, R)$, according to equation (6). Similarly, $w'(Y) = w(Y - \tilde{Y} + R) + D(\tilde{Y}, R)$ for some $\tilde{Y} \subseteq Y$ with $\left| \tilde{Y} \right| = |R|$.

According to the induction hypothesis, for $X - \tilde{X} + R$ and $Y - \tilde{Y} + R$, there exists a bijection $g : (X - \tilde{X} + R) \rightarrow (Y - \tilde{Y} + R)$ that satisfies the property GQ. We can further assume, from the preceding claim, that $g(r) = r$ for all $r \in R$.

Define the bijection $g' : X \rightarrow Y$:

Case $x \notin \tilde{X} : g'(x) := g(x)$,
Case $x = \tilde{x} \in \tilde{X} : g'(\tilde{x}) := g(\tilde{y})$, where $\tilde{y}$ is determined in the following way.

Let $M_{\tilde{X}}$ and $M_{\tilde{Y}}$ be the two minimum weight matchings whose values are $D(\tilde{X}, R)$ and $D(\tilde{Y}, R)$, respectively. Each point $r \in R$ is mapped by $M_{\tilde{X}}$ to some point $\tilde{x} \in \tilde{X}$ and by $M_{\tilde{Y}}$ to some point $\tilde{y} \in \tilde{Y}$.

To prove that w' satisfies the property GQ, consider any partition (X_1, X_2) of X. This partition implies a partition $(\tilde{X}_1, \tilde{X}_2)$ of $\tilde{X}$, where
$\tilde{X}_1 = \{x \in X | x \in X_1 \cap \tilde{X}\}$ and $\tilde{X}_2 = \{x \in X | x \in X_2 \cap \tilde{X}\}$.
Then, $X - \tilde{X} + R = X_1 - \tilde{X}_1 + X_2 - \tilde{X}_2 + R$ follows.

$$w'(X) + w'(Y) = w(X - \tilde{X} + R) + w(Y - \tilde{Y} + R) + D(\tilde{X}, R) + D(\tilde{Y}, R)$$

$$\geq w(X_1 - \tilde{X}_1 + g(X_2 - \tilde{X}_2 + R)) + w(g(X_1 - \tilde{X}_1) + X_2 - \tilde{X}_2 + R)$$
$$+ D(\tilde{X}, R) + D(\tilde{Y}, R),$$
by the induction hypothesis;

$$\geq w(X_1 - \tilde{X}_1 + g'(X_2) - g'(\tilde{X}_2) + R)$$
$$+ w(g'(X_1) - g'(\tilde{X}_1) + X_2 - \tilde{X}_2 + R) + D(\tilde{X}, R) + D(\tilde{Y}, R),$$
since $g(X_2 - \tilde{X}_2 + R) = g(X_2 - \tilde{X}_2) + g(R) = g'(X_2) - g'(\tilde{X}_2) + R$,

$$\geq w(X_1 + g'(X_2) - \tilde{X}_1 - g'(\tilde{X}_2) + R) + D(\tilde{X}_1, R^{(1)}) + D(g'(\tilde{X}_2), R^{(2)})$$
$$+ w(g'(X_1) + X_2 - g'(\tilde{X}_1) - \tilde{X}_2 + R) + D(g'(\tilde{X}_1), R^{(1)}) + D(\tilde{X}_2, R^{(2)}),$$
where $D(\tilde{X}, R) = D(\tilde{X}_1, R^{(1)}) + D(\tilde{X}_2, R^{(2)})$,
$D(\tilde{Y}, R) = D(\tilde{Y}_1, R^{(1)}) + D(\tilde{Y}_2, R^{(2)})$ with $\tilde{Y}_1 = g'(\tilde{X}_1)$,
$\tilde{Y}_2 = g'(\tilde{X}_2)$ according to the corresponding
partitionings of the matchings $M_{\tilde{X}, R}, M_{\tilde{Y}, R}$;

$$\geq w(X_1 + g'(X_2)) + w(X_2 + g'(X_1)),$$
using $D(\tilde{X}_1, R^{(1)}) + D(g'(\tilde{X}_2), R^{(2)}) \geq D(\tilde{X}_1 + g'(\tilde{X}_2), R)$,
$D(g'(\tilde{X}_1), R^{(1)}) + D(\tilde{X}_2, R^{(2)}) \geq D(g'(\tilde{X}_1) + \tilde{X}_2, R)$ and (5).

In case of the scarcity-situation there exists some $\tilde{X} \subseteq R$ with $\left|\tilde{X}\right| = k$, so that $w'(X) = w(\tilde{X}) + D(\tilde{X}, X)$, according to equation (10).

Similarly, $w'(Y) = w(\tilde{Y}) + D(\tilde{Y}, Y)$ for some $\tilde{Y} \subseteq R$ with $\left|\tilde{Y}\right| = k$.

According to the induction hypothesis, for $\tilde{X}$ and $\tilde{Y}$, there exists a bijection $g : \tilde{X} \rightarrow \tilde{Y}$ that satisfies the property GQ. We can further assume, from the preceding claim, that $g(r) = r$ for all $r \in \tilde{X} \cap \tilde{Y}$.

Define the bijection $g' : X \rightarrow Y$ in the following way:

Let $M_{\tilde{X}, X}$ and $M_{\tilde{Y}, Y}$ be two minimum weight matchings whose values are $D(\tilde{X}, X)$ and $D(\tilde{Y}, Y)$, respectively.

Set $g'(x) = y$ $(x \in X, y \in Y)$, if x is mapped by $M_{\tilde{X}, X}$ to $\tilde{x} \in \tilde{X}$, y is mapped by $M_{\tilde{Y}, Y}$ to $\tilde{y} \in \tilde{Y}$ and $g(\tilde{x}) = \tilde{y}$.

To prove that w' satisfies the property GQ, consider any partition (X_1, X_2) of X. This partition leads to a partition $(\tilde{X}_1, \tilde{X}_2)$ of $\tilde{X}$ by means of the matching $M_{\tilde{X}, X}$, where, in more detail, X_1 is mapped to $\tilde{X}_1$ by $M_{\tilde{X}, X}$.

$$w'(X) + w'(Y) = w(\tilde{X}) + w(\tilde{Y}) + D(\tilde{X}, X) + D(\tilde{Y}, Y)$$
$$\geq w(\tilde{X}_1 + g(\tilde{X}_2)) + w(g(\tilde{X}_1) + \tilde{X}_2)$$
$$+ D(\tilde{X}_1, X_1) + D(\tilde{X}_2, X_2) + D(\tilde{Y}_1, Y_1) + D(\tilde{Y}_2, Y_2), \qquad \text{by}$$

the induction hypothesis

$$\text{and where } g(\tilde{X}_2) =: \tilde{Y}_2, \; g(\tilde{X}_1) =: \tilde{Y}_1;$$
$$\geq w(\tilde{X}_1 + \tilde{Y}_2) + D(\tilde{X}_1, X_1) + D(\tilde{Y}_2, Y_2)$$
$$+ w(\tilde{Y}_1 + \tilde{X}_2) + D(\tilde{X}_2, X_2) + D(\tilde{Y}_1, Y_1)$$
$$\geq w(X_1 + Y_2) + w(Y_1 + X_2),$$
$$\text{using } D(\tilde{X}_1, X_1) + D(\tilde{Y}_2, Y_2) \geq D(\tilde{X}_1 + \tilde{Y}_2, X_1 + Y_2),$$
$$D(\tilde{X}_2, X_2) + D(\tilde{Y}_1, Y_1) \geq D(\tilde{X}_2 + \tilde{Y}_1 X_2 + Y_1) \text{ and (5)},$$
$$\text{where } Y_1 = g'(X_1), Y_2 = g'(X_2). \qquad \square$$

C Example

Let $k = 5$ and let the metric space consist of 8 points: $M = \{p_1, p_2, \cdots, p_8\}$. The distances between the points are listed in the following matrix

$$\begin{pmatrix} 0 & 41 & 31 & 44 & 43 & 45 & 29 & 27 \\ 41 & 0 & 51 & 52 & 37 & 53 & 42 & 35 \\ 31 & 51 & 0 & 53 & 46 & 38 & 47 & 48 \\ 44 & 52 & 53 & 0 & 40 & 36 & 39 & 28 \\ 43 & 37 & 46 & 40 & 0 & 30 & 32 & 50 \\ 45 & 53 & 38 & 36 & 30 & 0 & 49 & 34 \\ 29 & 42 & 47 & 39 & 32 & 49 & 0 & 55 \\ 27 & 35 & 48 & 28 & 50 & 34 & 55 & 0 \end{pmatrix}.$$

The initial configuration is given by $S^0 = \{p_1, p_2, p_3, p_4, p_5\}$ and the request sequence

$\sigma = R^1, R^2, \cdots, R^5$ by $R^1 = \{p_6\}, R^2 = \{p_7\}, R^3 = \{p_8\}, R^4 = \{p_6\},$
$R^5 = \{p_4, p_5\}$.

Since $|R^1| = \cdots = |R^4| = 1$, the algorithms WFA and compound WFA yield the same costs and the same configurations after servicing subsequences $\sigma_t, \ t = 1, 2, 3, 4$. More detailed, $S^1 = \{p_1, p_2, p_3, p_4, p_6\}$, $S^2 = \{p_2, p_3, p_4, p_6, p_7\}$, $S^3 = S^4 = \{p_2, p_3, p_6, p_7, p_8\}$. The costs after servicing the subsequence σ_4 are equal to 87.

Finally, WFA (from Section 4) leads to $S^5 = \{p_4, p_5, p_6, p_7, p_8\}$ and cost equal to 177. The compound WFA is based on a surrogate problem with requests
$\bar{R}^{1,1} = \{p_4\}, \bar{R}^{1,2} = \{p_5\}, \ldots$ in step 5. We get $S'^{1,1} = \{p_2, p_3, p_4, p_6, p_7\}$,
$S'^{1,2} = \{p_3, p_4, p_5, p_6, p_7\}$. Since $R^5 \subseteq S'^{1,2}$, $S'^{1,2} = \{p_3, p_4, p_5, p_6, p_7\}$ is the end configuration of compound WFA with cost equal to 152. Thus, the compound WFA algorithm yields a better result than the WFA algorithm for this example.

A computer was used for the calculations. In addition, we give some selected values of work functions: $w_4(\{p_4, p_5, p_6, p_7, p_8\}) = 102$, $w_4(\{p_3, p_4, p_5, p_6, p_7\}) = 128$, $w_4(\{p_2, p_3, p_4, p_6, p_7\}) = 115$ and $w_4(\{p_2, p_4, p_6, p_7, p_8\}) = 97$. We can observe that $w_4(S^5) + D(S^4, S^5) < w_4(S'^{1,2}) + D(S'^{1,2}, S^4)$ and
$w_4(S'^{1,1}) + D(S^4, S'^{1,1}) < w_4(\{p_2, p_4, p_6, p_7, p_8\}) + D(S^4, \{p_2, p_4, p_6, p_7, p_8\})$, for example. This implies the different results of WFA and compound WFA.

Author's information

R. Hildenbrandt received a PhD degree as well a doctor habilitatus degree from Ilmenau Technical University and works now as a Privatdozent in the Optimization Department on stochastic dynamic programming and online optimization.