Available online at www.sciencedirect.com

ScienceDirect

Procedia CIRP 41 (2016) 460 – 465

www.elsevier.com/locate/procedia

48th CIRP Conference on MANUFACTURING SYSTEMS - CIRP CMS 2015

Simultaneous Workload Allocation and Capacity Dimensioning for Distributed Production Control

Henning Blunck[a]*, Dieter Armbruster[b], Julia Bendul[a]

[a] School of Mathematics and Logistics, Jacobs University, Campus Ring 1, 28759, Bremen, Germany
[b] School of Mathematical and Statistical Sciences, Arizona State University, P.O. Box 871804, Tempe, AZ 85287-1804

* Corresponding author. Tel.: +49 421 200 3075; fax: +49 421 200 3078. *E-mail address:* h.blunck@jacobs-university.de

Abstract

Capacity dimensioning in production systems is an important task within strategic and tactical production planning which impacts system cost and performance. Traditionally capacity demand at each worksystem is determined from standard operating processes and estimated production flow rates, accounting for a desired level of utilization or required throughput times. However, for distributed production control systems, the flows across multiple possible production paths are not known a priori. In this contribution, we use methods from algorithmic game-theory and traffic-modeling to predict the flows, and hence capacity demand across worksystems, based on the available production paths and desired output rates, assuming non-cooperative agents with global information. We propose an iterative algorithm that converges simultaneously to a feasible capacity distribution and a flow distribution over multiple paths that satisfies Wardrop's first principle. We demonstrate our method on models of real-world production networks.

Peer-review under responsibility of the scientific committee of 48th CIRP Conference on MANUFACTURING SYSTEMS - CIRP CMS 2015

Keywords: Agent Based Manufacturing Control, Capacity Dimensioning, Resource Requirements Problem, Algorithmic Game Theory

1. Introduction

After deciding on the planned production program, a production planner has to decide on (1) the production processes to use and allocate expected production flow to each of them, (2) calculate the capacity demand at each worksystem, and (3) set resource capacities, so as to attain a target utilization rate that balances high utilization and short cycle times [c.f. 1, Ch. 3.2].

Usually these steps are executed sequentially: A fixed flow distribution on one or many process paths for a given end product is given, assuming a fixed distribution of flow between the alternative paths before the capacity is dimensioned. However such a setup is not compatible with distributed control settings, where routing decisions are usually delayed and left to the product at run-time [2]. Without reliable information on expected flow distributions, the production designer cannot production capacities properly and runs the risk of observing unexpected levels of utilization and lead times, making production systems with distributed control less predictable and hence hindering their acceptance in industry [2, 3].

To overcome this problem, we extend a method from traffic modeling to predict the flow of non-cooperating agents in networks with alternative process paths. We devise an iterative algorithm that simultaneously sizes the capacities and determines the flow of agents.

2. Existing Approaches

Traditional manufacturing system design frameworks consider the steps of process path selection (and distribution of flow among them) separate from the subsequent tasks of capacity requirements calculation and capacity dimensioning [c.f. 1, Ch. 3.2]. Given a known set of production paths and

Peer-review under responsibility of the scientific committee of 48th CIRP Conference on MANUFACTURING SYSTEMS - CIRP CMS 2015
doi:10.1016/j.procir.2015.12.117

flow distributions among them, the machine requirements and capacity dimensioning problems have been studied since the 1950s (see [4] for a review of early work). Typically the number of required machines per operation is calculated by dividing the capacity demand (calculated as the product of deterministic or expected flow and processing time) by the total working time in the same time-period [4]. Subsequent research aimed at formulating optimization models that (considering, capacity, floor space, cost, ...) calculated the optimal number of machines, still assuming fixed flow distribution [5].

With the rise of modern manufacturing design paradigms, such as Flexible Manufacturing Systems (FMS), researchers have tried to take into account alternative process paths in the decision making process. An integrated approach was proposed in [6] in the form of an optimization problem based on flow independent waiting times. In FMS design, "processing capacity issues" have largely been tackled through queuing networks and related models [7], with some contributions also including the flow distribution over path alternatives in the optimization search space: Both Tetzlaff [8] and Lee et al. [9] discuss the cost-minimal FMS configuration capable of fulfilling production requirements. While accounting for queuing effects, they do not consider autonomous (agent-controlled) products. For a review, the reader is referred to [10].

Beside analytical and optimization approaches, simulation models can be applied for the resource requirements problem, however only for rather small systems, where the parameter space is feasibly explorable [11, Ch. 5.3].

The concept of interaction between intelligent products and adjustable machine capacity has been experimentally and analytically investigated in [12], focusing on short term capacity adjustment rather than production system design. Previous simulation experiments where capacity dimensioning had to be made in the context of distributed production usually focused on parallel worksystems [e.g. 13] (or linear sequences thereof [14]), where the capacity problem can easily be solved (assign equal capacity to each parallel server).

Differing from previous work, our approach accounts for non-cooperative decision making by autonomous product-agents in arbitrary networks (see also Sec. 7), considering flow dependent delays (queueing effects). By solving a game-theoretic equilibrium problem we forgo computationally expensive simulation experiments and hence can operate more quickly and comprehensively on the solution space than the optimization approaches discussed above.

3. Prediction of Traffic Flows through Game-Theory

Predicting the routing behavior of individual, non-cooperative agents is known in traffic-modeling as the Traffic Assignment (TA) problem [15]. Game-theoretic approaches to solve the TA problem find a Nash equilibrium which in this context is known as Wardrop's first principle [16]. Wardrop states that for individual agents choosing one of several alternative routes, the steady state distribution will be such that

> "The journey times in all routes actually used are equal and less than those which would be experienced by a single vehicle on any unused route." [16]

The usual assumptions about Nash Equilibiria translate into three important consequences of Wardrop's principle: (i) The agents require perfect information about the current state of the system - in particular the waiting times at all relevant worksystems - in order to compare the latency along all possible paths, (ii) the traffic is assumed to be nonatomic (i.e. the size of one unit of flow is marginal as compared to the overall flow), and (iii) the cost equilibrium only holds for paths with positive flow (i.e. there can be unused paths).

The assumption of perfect information is an issue in the context of distributed decision making systems, for which a local information horizon is commonly considered (c.f. e.g. [2, 17] and [18, Ch. 1.1]). It should be noted though that some methods of distributed control (e.g. pheromone-based approaches) lead to a buildup of "global information" through dropped pheromones accessible locally [c.f. 19, 20, 21]. Papadimitriou and Valiant [22] present an approach to flows distributions satisfying Wardrop's first principle, where agents use local information only. However, they do not present an algorithm to calculate them for arbitrary network structures.

The nonatomic flow assumption will barely hold for the actual operation of the factory (or road network) but is a tribute to the notion of long-term steady state distributions and a necessary assumption in Wardrop's principle. Also in the discussion of distributed control, the nonatomic flow assumption is not new [c.f. e.g. 12, 21].

The third consequence implies that even for linear cost functions, the problem is not a linear problem, but rather a Mixed Integer Linear Programming problem where a subset of alternative paths has to be selected. Hence the problem cannot be solved as a set of equations, but naturally entails a notion of subset selection which could require enumeration of path combinations at prohibitive computational cost [23].

Fortunately, with the Method of Successive Averages (MSA) there exists a widely used heuristic to find Nash equilibria in traffic assignment problems [24]. The idea is the following: Given a flow distribution $\overrightarrow{f^{i}}$, calculate the "cost" of each path alternative. Create a distribution $\vec{h}$ by assigning all flow to the "cheapest" alternative and calculate the new flow distribution $\overrightarrow{f^{i+1}}$ as the weighted average of old and new flow. Sheffy & Powell [25] proposed, using a particular weighting for the averages for which they showed in [23] that the process, given increasing convex cost functions (as function of flow), will converge to a Nash equilibrium.

The requirement on the cost functions (waiting time on a road in the TA problem, throughput time at a machine in the production equivalent) to be non-decreasing with flow, holds for all typical clearing functions [26]. So our model poses little restrictions on e.g. the queuing model class incorporated, allowing for example the incorporation of variability in arrival and departure processes or machine breakdowns, by using the respective cycle time approximations.

Algorithm 1 (Capacity Dimensioning for Agent-Based Production Systems).

1. *Calculate initial flow distribution* $\vec{f^0} = \vec{f}_{initial}$.
2. *Calculate capacity distribution* $\vec{\mu} = \frac{\Phi \cdot \vec{f^0}}{u^*}$ *where* u^* *is the target utilization.*
3. *Apply Method of Successive Averages:*
   (a) *Set* $i := 0$.
   (b) *Calculate cost per path* $\vec{c_p}(f^i)$.
   (c) *Calculate new flow distribution* $\vec{h}$ *as follows, using an "All-Or-Nothing Approach": For every set of alternative paths, assign all flow to the path with the lowest cost in* $\vec{c_p}(f^i)$.
   (d) *Calculate* $\vec{f}^{i+1} = (1-\lambda)\cdot\vec{f^i} + \lambda\cdot\vec{h}$ *with* $\lambda = \frac{i}{i+1}$.
   (e) *Set* $i := i + 1$.
   (f) *Repeat steps 3b - 3e until convergence criterion is met.*
4. *Calculate capacity utilization, given capacities* $\vec{\mu}$ *and flow* $\vec{f^i}$ *as* $\vec{u} = \frac{\Phi \cdot \vec{f^i}}{\vec{\mu}}$.
5. *Repeat steps 2 - 4 with* $\vec{f^0} := \vec{f^i}$ *until convergence criterion is met.*

4. Proposed Iterative Approach to Capacity Dimensioning

The proposed algorithm, shown in Alg. 1, wraps the (already iterative) MSA algorithm (Step 3) into another loop in which the predicted flow distribution (the outcome of the MSA algorithm), given the current capacity distribution, is then used to calculate utilization rates per worksystem (Step 4). Subsequently, unless a set convergence criterion is met, worksystem capacities are adjusted such that, given the calculated flow, all worksystems would have the desired capacity utilization (Step 2). The algorithm end, if the flow distribution resulting from the MSA run is "close enough" to the distribution assumed in the last round of capacity adjustments and hence the actual capacity utilization levels are close to the desired value. In other words: The flow resulting from non-cooperative agents sufficiently matches the flow expected during the capacity dimensioning process.

To generate an initial flow distribution (Step 1), we use the "Equal Share Assignment" rule, proposed in [27] (see also [15, Sec. 4.2.1]), which distributes flow evenly among all alternative paths. The following notation is based on [15]: Let $\mathcal{M}$ be the set of all worksystems (with length $|\mathcal{M}|$ = m) and $\vec{\mu}$ the capacity distribution among them. Let $\mathcal{R}$ be the set of all possible paths in the systems ($|\mathcal{R}| = \rho$). Then $\Phi \in \mathbb{R}^{m\times\rho}$ is the flow-path incidence matrix, where elements $\varphi_{i,j}$ indicate the number of times, worksystem i is visited by path j. Let $\vec{f}$ be a flow distribution over $\mathcal{R}$, then $\lambda = \Phi \cdot \vec{f}$ gives the capacity demand (flow) for all worksystems.

Each worksystem $i \in \mathcal{M}$ has a (convex) cost (i.e. cycle time) function $c_i(\lambda_i)$, yielding a per-worksystem cost vector (given the current flow distribution and capacity levels) $\overrightarrow{c_m}$, from which cost per path ($\overrightarrow{c_p}$) can be calculated as $\overrightarrow{c_p} = \Phi^T \cdot \overrightarrow{c_m}$.

5. Experimental Setup

To show the applicability even on large scale production networks, we demonstrate the methods on networks derived from actual production system's feedback data. We use the same datasets as investigated in [28] (c.f. Tab. 1), except for Company D, where the feedback data was not available in a suitable format. We seek to create a capacity distribution such that autonomous controlled product flow would result in an 80% target utilization. The following assumptions are necessary to transform the feedback data into networks of alternative production paths. They do not do justice to the intricacies of the actual production systems, but are deemed acceptable for a first proof-of-concept.

- All process paths with the same first and last worksystem are considered alternative.
- The total flow for one origin-destination (OD) relationship is the sum of products (each given weight 1) recorded over all paths between this origin and destination.
- The cost function $c(\lambda) = 1/(\mu - \lambda)$ represents the cycle time of an M/M/1 queueing system where μ is the capacity and λ is the flow [11, Ch. 3.2]. However, to ensure convexity for utilization levels > 1 and to bar products from disestablished worksystems, we define the worksystem cost $c_i(\lambda_i)$ piecewise as follows:

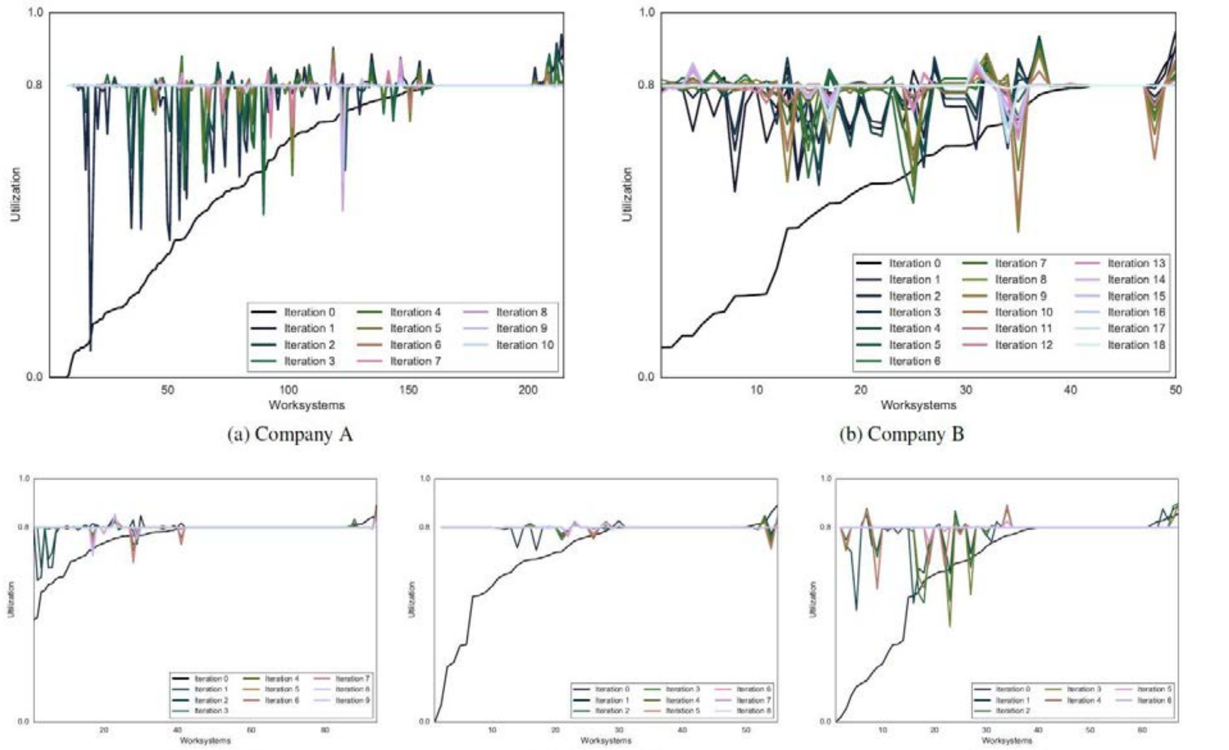
$$c_i(\lambda_i)x = \begin{cases} 10^{20} & if\ \mu_i = 0 \\ \dfrac{1}{\mu - \lambda} & if\ \mu_i \neq 0\ \wedge\ \lambda_i \leq 0.98\mu_i \\ c(0.98\mu_i) + (\lambda_i - 0.98\mu_i)\cdot c'(0.98\mu_i) & else \end{cases}$$

- The processing time is one time unit for every item on every machine.

After the initial network generation, we remove all pareto-dominated paths. We assume a path a to be pareto-dominated by path b *if and only if* a is longer and the set of worksystems in a is a superset of those in b. Note that this filtering of process alternatives improves performance (c.f. Fig. 1) but is not necessary, since the MAS approach filters out inefficient paths automatically.

Company	Type	Worksystems	OD-Pairs	Undom. Paths	Paths/OD-Pair	Req. Iterations
A	Job-Shop	215 (220)	1190	2114	1.78	11
B	Job-Shop	50 (51)	182	376	2.07	19
C	Job-Shop	94 (98)	690	853	1.24	10
E	Job-Shop	55 (57)	185	263	1.42	9
F	Customizing	67 (87)	204	342	1.67	7

Table 1: Datasets analyzed in this publication. Originally investigated and named in [28]. Worksystems column shows number of worksystems on undominated paths and the total number in brackets.



(a) Company A (b) Company B

(c) Company C (d) Company E (e) Company F

Figure 1: Development of capacity utilization over iterations on networks from [28]. Worksystems are ordered by the utilization attained with after initial round (black line). For the MSA, a maximum flow deviation of 1% was chosen as convergence criterion. The overall process was terminated, once all worksystem utilization levels were within a ±2% window around the target utilization rate (80%).

6. Results

The experimental results for these (suitably transformed) real production networks are shown in Fig. 1. Worksystems are ordered by the capacity utilization attained after the first round (black line) where capacities were dimensioned according to an "equal-share" flow-distribution. We find:

- All test-instances eventually converge to a scenario where the assumed flow (based on which the capacities were dimensioned) and the actual flow (using MSA to assign flow) matches and all worksystems with non-zero flow are within ±2% of the target utilization.
- The algorithm correctly eliminates dominated flows: Fig. 1a shows an instance where the iterated application of agent-based control leads to the elimination of worksystems (all paths that crossed the worksystems in the original equal share flow assignment have been found dominated) and hence the final capacity for these worksystems is 0.
- The number of iterations required and the associated computational effort has been limited so far (no more than 19 iterations or about one hour of computation time in one instance). The number of iterations required increases with the number of path alternatives (c.f. Fig. 2) and the tolerance window for attained utilization levels. The computational effort per round is primarily affected by the convergence criteria set for the MSA algorithm. The worksystems that show a constant 80% utilization level throughout all iterations are those which are only found on OD-pairs without alternatives. For these worksystems, we can calculate the required capacity demand and hence the necessary capacity in the known fashion (c.f. Sec. 2). Instances with many such "easy" worksystems (Fig. 1c through 1e) are unsurprisingly the quickest to converge.

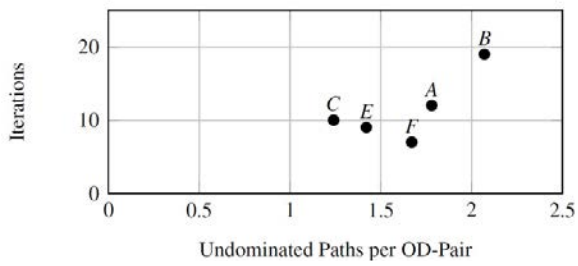


Figure 2: Required iterations as function of undominated paths per Origin-Destination pair for networks from Tab. 1.

7. Discussion and Possible Extensions

We have proposed and demonstrated on realistic production networks, an algorithm to dimension capacities in the presence of agent-controlled, autonomous products. The algorithm simultaneously yields a capacity distribution and steers non-cooperative, minimal throughput time seeking agents towards a flow distribution that yields the expected utilization (and hence cycle time) levels. It thus reduces the unpredictability of autonomous control in the design and implementation phase of manufacturing systems. The approach significantly extends previous research which assumed central control over (or even fixed) distribution of flow over alternative production paths.

While demonstrated here for simple paths, our algorithm holds for general network structures, including multigraphs such as assembly and disassembly networks, as long as the set of alternative paths is enumerable and the flow experienced at a worksystem, given a unit-flow across a particular path, is known.

The obvious next step is a validation through discrete event simulation. A fluid approach would be easier*, however it would not be sufficient to validate the nonatomic flow approximation.

The game-theoretic founding of the applied concept allows for further, game-theory inspired investigations into the performance of autonomous control architectures in production environments: It is well known that the Nash-Equilibrium (NE), i.e. the steady state achieved by the interaction of selfish agents, may have higher social cost than the social optimum [29]. The social-optimum flow distribution for convex cost functions can be attained by finding the Nash-Equilibrium when the worksystem cost functions are replaced by the marginal cost function [30]

$$\frac{\delta}{\delta f}(c_a(f)\cdot f).$$

The ratio of NE-cost and optimal cost is called *"Price of Anarchy"* [31]. The identification of networks non socially-optimal flow Nash-Equilibria has so far only been tackled in form of an optimization problem in [32] and capacity distributions that guarantee to exhibit no cost of anarchy can only be guaranteed for very simple setups with parallel processors [33]. Little is known about the (network-structural) relative aptitude towards agent-based control. Our initial investigations did not show significant prices of anarchy, however, since those only arise for small ranges of flow [22], more investigations are necessary.

* Since it is deterministic and does not rely on the processing of future event lists

References

[1] M. Schenk, S. Wirth, E. Müller, Factory Planning Manual - Situation-Driven Production Facility Planning, 1st Edition, Springer Science + Business Media, 2010.

[2] D. Trentesaux, Distributed control of production systems, Engineering Applications of Artificial Intelligence 22 (7) (2009) 971 – 978, distributed Control of Production Systems.

[3] V. Marik, D. McFarlane, Industrial adoption of agent-based technologies, Intelligent Systems, IEEE 20 (1) (2005) 27–35.

[4] D. M. Miller, R. P. Davis, The machine requirements problem, International Journal of Production Research 15 (2) (1977) 219–231.

[5] A. Kusiak, The production equipment requirements problem., International Journal of Production Research 25 (3) (1987) 319.

[6] J. F. Bard, T. A. Feo, An algorithm for the manufacturing equipment selection problem, IIE Transactions 23 (1) (1991) 83–92.

[7] P. Solot, M. van Vliet, Analytical models for fms design optimization: A survey, International Journal of Flexible Manufacturing Systems 6 (3) (1994) 209–233.

[8] U. A. Tetzlaff, A model for the minimum cost configuration problem in flexible manufacturing systems, International Journal of Flexible Manufacturing Systems 7 (2) (1995) 127–146.

[9] H. Lee, M. Srinivasan, C. Yano, The optimal configuration and workload allocation problem in flexible manufacturing systems, International Journal of Flexible Manufacturing Systems 3 (3-4) (1991) 213–230.

[10] H. F. Lee, M. M. Srinivasan, C. A. Yano, A framework for capacity planning and machine configuration in flexible assembly systems, International Journal of Flexible Manufacturing Systems 18 (4) (2006) 239–268.

[11] G. Chryssolouris, Manufacturing systems: theory and practice, 2nd Edition, Springer-Verlag, 2006.

[12] O. Jeken, N. Duffie, K. Windt, H. Blunck, A. Chehade, H. Rekersbrink, Dynamics of autonomously acting products and work systems in production and assembly, CIRP Journal of Manufacturing Science and Technology 5 (4) (2012) 267 – 275, special issue from 44th CIRP Conference on Manufacturing Systems.

[13] B. Scholz-Reiter, M. Görges, T. Philipp, Autonomously controlled production systems – influence of autonomous control level on logistic performance, CIRP Annals - Manufacturing Technology 58 (1) (2009) 395 – 398.

[14] B. Scholz-Reiter, M. Freitag, C. de Beer, T. Jagalski, Modelling dynamics of autonomous logistic processes: Discrete-event versus continuous approaches, {CIRP} Annals - Manufacturing Technology 54 (1) (2005) 413 – 416.

[15] A.Raith, J.Wang, M.Ehrgott, S.Mitchell, Solvingmulti-objective traffic assignment, Annals of Operations Research 222 (1) (2014) 483–516.

[16] J. G. Wardrop, Some theoretical aspects of road traffic research, ICE Proceedings: Engineering Divisions 1 (3) (1952) 325–362.

[17] P. Leitão, Agent-based distributed manufacturing control: A state-of-the-art survey, Engineering Applications of Artificial Intelligence 22 (7) (2009) 979 – 991, distributed Control of Production Systems.

[18] C. Schneeweiß, Distributed decision making, 2nd Edition, Springer, 2003.
[19] P. Peeters, H. V. Brussel, P. Valckenaers, J. Wyns, L. Bongaerts, M. Kollingbaum, T. Heikkil, Pheromone based emergent shop floor control system for flexible flow shops, Artificial Intelligence in Engineering 15 (4) (2001) 343 – 352, methodology of Emergent Sythesis.
[20] H.VanDykeParunak, “go to the ant”: Engineering principles from natural multi-agent systems, Annals of Operations Research 75 (0) (1997) 69–101.
[21] D. Armbruster, C. de Beer, M. Freitag, T. Jagalski, C. Ringhofer, Autonomous control of production networks using a pheromone approach, Physica A: Statistical Mechanics and its Applications 363 (1) (2006) 104 – 114, information and Material Flows in Complex Networks: Information and Material Flows in Complex Networks.
[22] C. Papadimitriou, G. Valiant, A new look at selfish routing, in: Innovations in Computer Science (ICS2010), 2010, pp. 178–187. URL http://conference.itcs.tsinghua.edu.cn/ICS2010/content/paper/Paper_15.pdf
[23] W. B. Powell, Y. Sheffi, The convergence of equilibrium algorithms with predetermined step sizes, Transportation Science 16 (1) (1982) 45–55.
[24] H. Sbayti, C.-C. Lu, H. S. Mahmassani, Efficient implementation of method of successive averages in simulation-based dynamic traffic assignment models for large-scale network applications, Transportation Research Record: Journal of the Transportation Research Board 2029.
[25] Y. Sheffi, W. Powell, A comparison of stochastic and deterministic traffic assignment over congested networks, Transportation Research Part B: Methodological 15 (1) (1981) 53 – 64.
[26] D. Armbruster, R. Uzsoy, Continuous Dynamic Models, Clearing Functions, and Discrete-Event Simulation in Aggregate Production Planning, 2012, Ch. 7, pp. 103–126.
[27] J.Y.Wang, A.Raith, M.Ehrgott, Tolling analysis with bi-objective traffic assignment, in: M. Ehrgott, B. Naujoks, T. J. Stewart, J. Wallenius (Eds.), Multiple Criteria Decision Making for Sustainable Energy and Transportation Systems, Vol. 634 of Lecture Notes in Economics and Mathematical Systems, Springer Berlin Heidelberg, 2010, pp. 117–129.
[28] T. Becker, M. Meyer, K. Windt, A manufacturing systems network model for the evaluation of complex manufacturing systems, International Journal of Productivity and Performance Management 63 (3) (2014) 324–340.
[29] P. Dubey, Inefficiency of nash equilibria, Mathematics of Operations Research 11 (1) (1986) 1–8.
[30] T. Roughgarden, Selfish routing and the price of anarchy, Vol. 174, MIT press Cambridge, 2005.
[31] E. Koutsoupias, C. Papadimitriou, Worst-case equilibria, Computer Science Review 3 (2) (2009) 65 – 69.
[32] J. Hagstrom, R. Abrams, Characterizing braess’s paradox for traffic networks, in: Intelligent Transportation Systems, 2001. Proceedings. 2001 IEEE, 2001, pp. 836–841.
[33] Y. Korilis, A. Lazar, A. Orda, Capacity allocation under noncooperative routing, Automatic Control, IEEE Transactions on 42 (3) (1997) 309–325.