



# Optimized Staffing Levels in a Ticketing Service System

Redha Bournas & Brian T. Eck

Global Technology Services, IBM Corporation



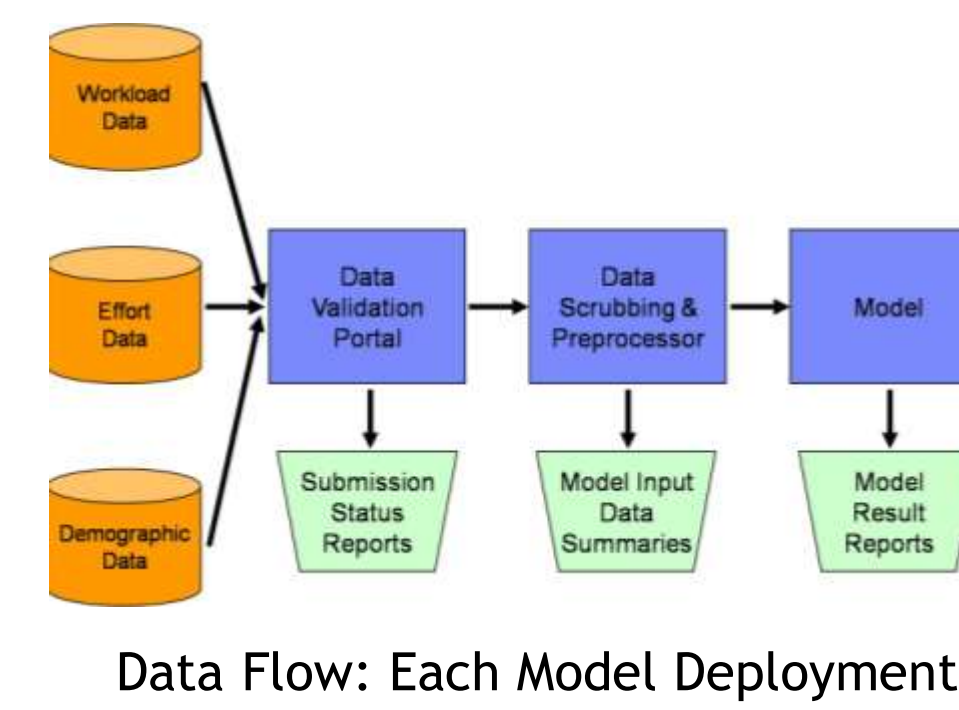
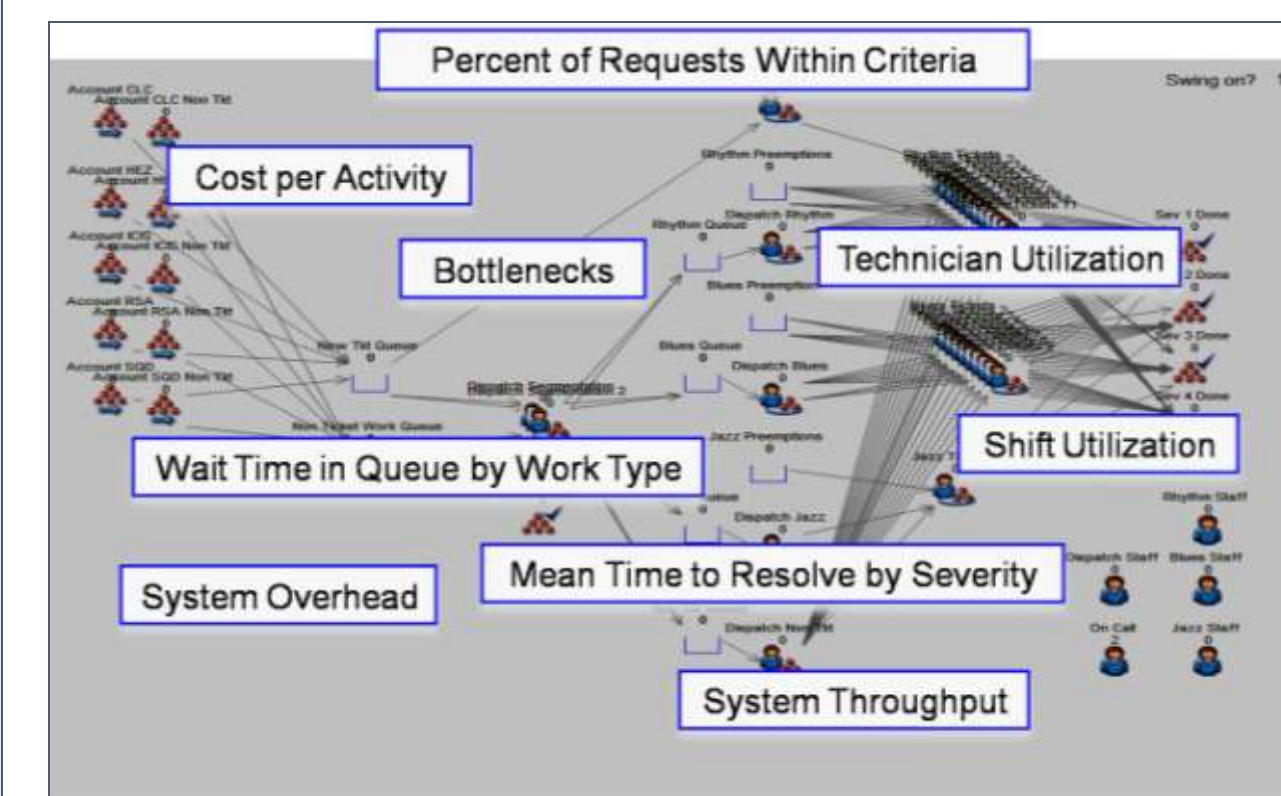
## BACKGROUND

**Q:** If you have thousands of teams delivering hundreds of services worldwide, ... How do you know that you have the correct staffing level in each team to meet your customers' needs?

**A:** You don't,

... Unless you can use the power of simulation to model those teams and determine the right staffing levels.

Simulation Models Give Us Insights into Many Team Performance Characteristics



Local search calls simulation to test feasibility and performance of each solution

### Optimization with Simulation

1. Construct base simulation model (used as evaluation function)

$$\min \sum_{i=1}^N \sum_{k=1}^7 \sum_{j=1}^M \sum_{l=1}^L \sum_{m=1}^M \sum_{n=1}^N \sum_{o=1}^O \sum_{p=1}^P \sum_{q=1}^Q \sum_{r=1}^R \sum_{s=1}^S \sum_{t=1}^T \sum_{u=1}^U \sum_{v=1}^V \sum_{w=1}^W \sum_{x=1}^X \sum_{y=1}^Y \sum_{z=1}^Z$$

2. Construct math formulation of the optimization problem (finding a min-cost staffing solution)

3. Translate (associate) with input spreadsheet variables, constraints, data

4. Build optimization "experiment" into AnyLogic with OptQuest engine

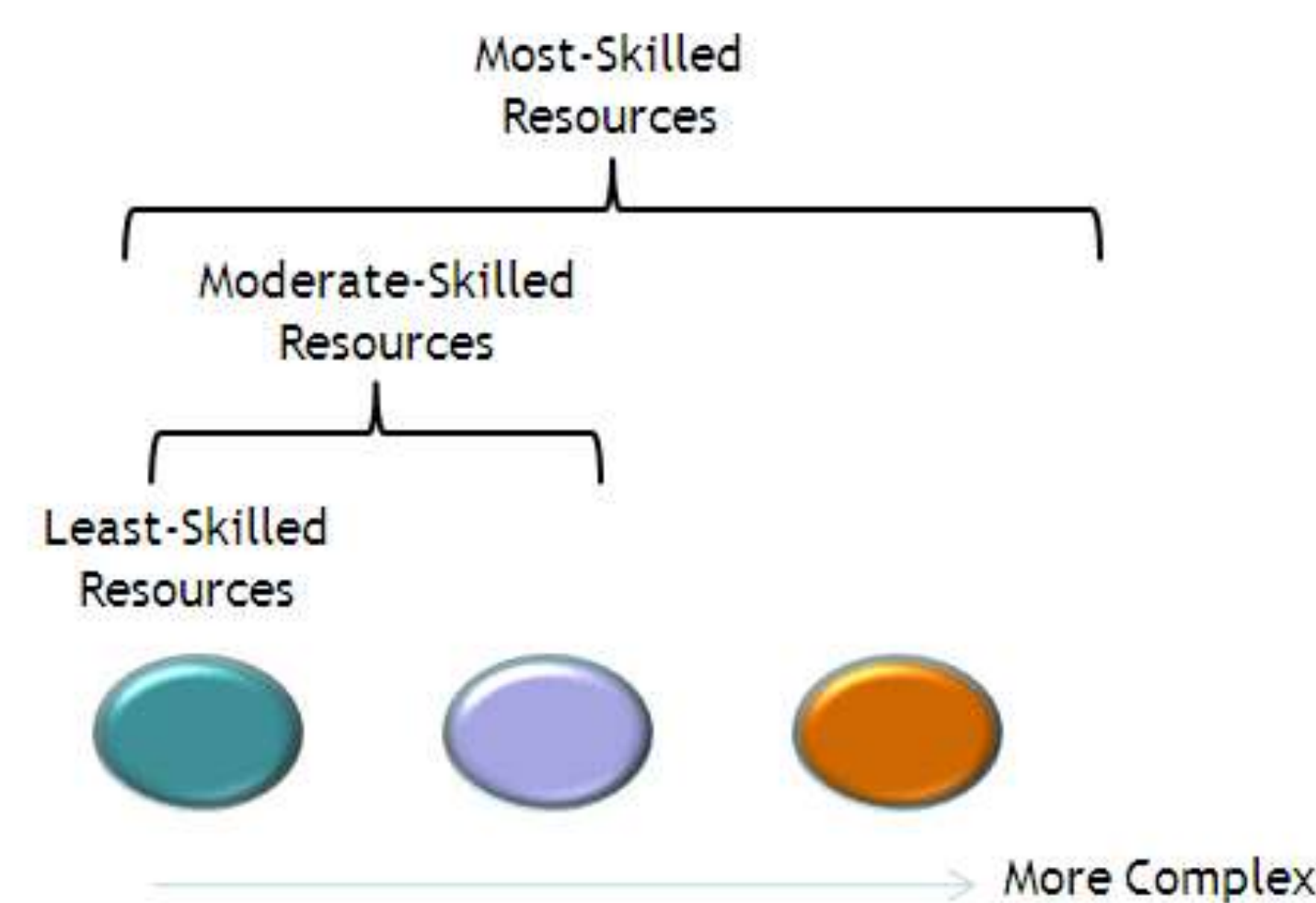


5. "Let 'er rip"

## Setting: Basics of Underlying Model

Complexity classes of service: For each, there is a corresponding pool of resources.

We model this as a queuing system with an allocated queue for each service complexity class.



Parameters of Model:

Ticket arrivals (nonhomogeneous Poisson), service times, dispatching policies, shift schedules, break logic, and swinging resources between complexity class queues.

Additional characteristics for tickets and resources can include accounts and technologies served (future extensions of model).

## Problem Statement

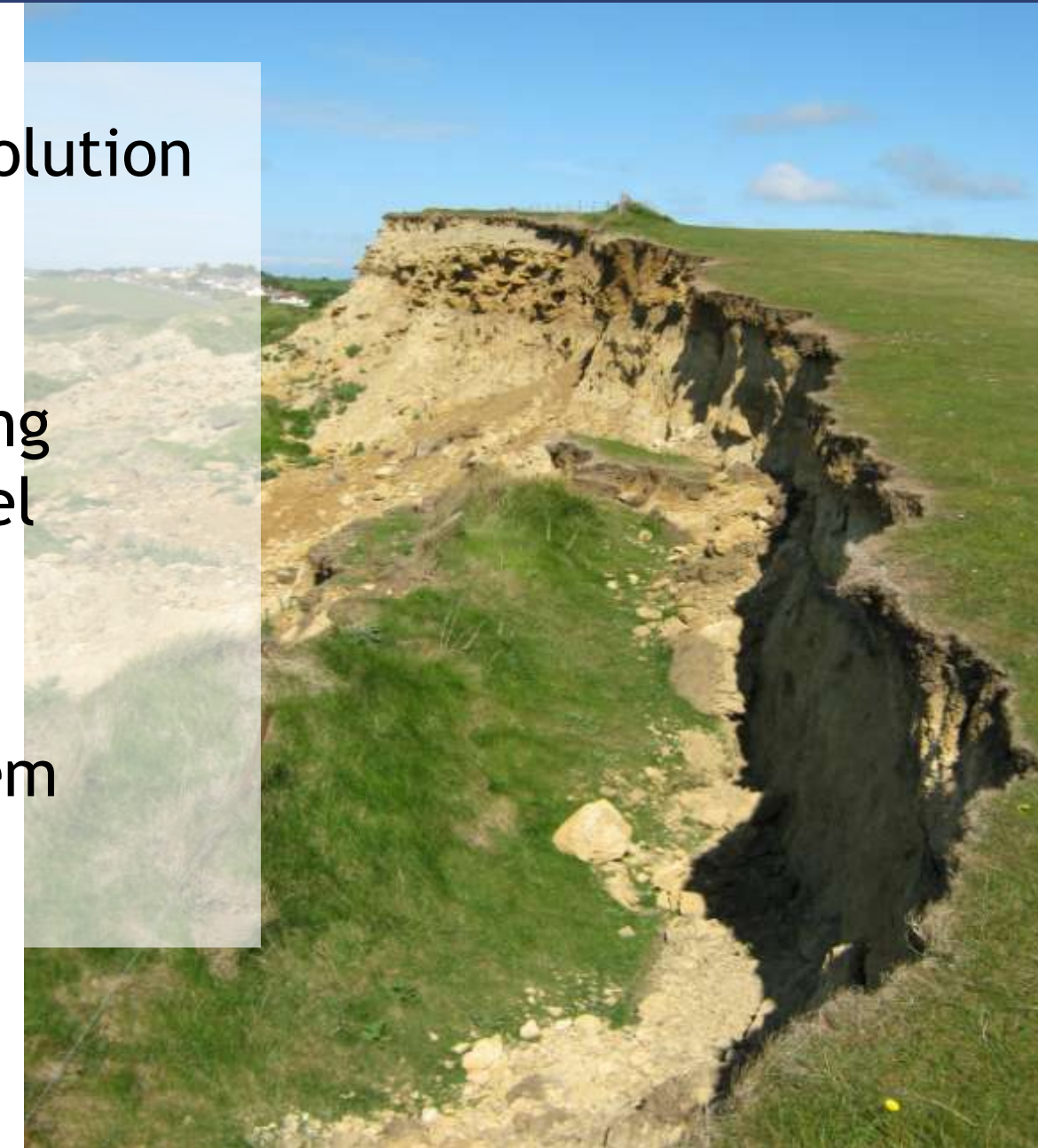
Determine the minimal number of pool resources, at each skill level, per shift, to achieve the resolution time service level agreements (SLA's) associated with incoming tickets from various accounts.

## Generic Local Search

Starting point is an initial feasible solution (provides an upper bound).

Evaluation is conducted by simulating current staffing against Service Level Agreement (SLA) attainment.

Practical Challenge: Detecting System Instability



Can we be smarter in our search?

## Guided Heuristic Search

Starting point is a lower bound and initial *infeasible* solution(s), which we improve in the direction of feasibility

Deriving stability equations can assist with both the initial solutions for this guided search, as well as prescreening unbounded iterations in the generic local search approach.

Initially we consider the simpler situation where there is a single service class.

## System Stability Equation

- $\lambda$ : Weekly average arrival rate of tickets for a given class
- $\mu$ : Hourly average service rate of tickets for that class
- $N$ : Number of shift schedules
- $d^{(i,k)}$ : Duration of shift schedule  $i$  in day  $k$  of the week
- $m_i$ : Number of service agents during shift schedule  $i$

$$\lambda < \mu \left( \sum_{i=1}^N m_i \sum_{k=1}^7 d^{(i,k)} \right)$$

## Single Service Class Problem

The following Integer Linear Program provides a lower bound on the problem objective, since it relaxes the SLA constraints.

$$\text{Minimize } \sum_{i=1}^N m_i$$

$$\text{subject to } \mu \left( \sum_{i=1}^N m_i \sum_{k=1}^7 d^{(i,k)} \right) > \lambda$$

$$m_i \geq 0, \text{ integer}$$

$$\text{Letting } j = \arg \max_{k=1, \dots, 7} \{ \sum_{i=1}^N d^{(i,k)} \}$$

an optimal solution to the IP is given by:

$$m_j^{(1)} = \lceil \lambda / \left( \mu \sum_{k=1}^7 d^{(j,k)} \right) \rceil$$

$$m_i^{(1)} = 0 \quad i \neq j (i=1, 2, \dots, N)$$

Observation:

This solution, while a viable lower bound on the original single class problem, may not be close to feasible with the added SLA constraints.

To construct a "closer to feasible" starting point, we add the constraint that the resources working within a shift, service the work arriving in their shift. While this is not a necessary condition for the optimal solution, it has an objective value not far from the lower bound just established.

Letting  $\lambda_i$  be the average arrival rate of tickets during the shift schedule  $i, i=1, 2, \dots, N$ , another lower bound for the optimal solution to the "per shift" problem is given by:

$$m_i^{(2)} = \lceil \lambda_i / \left( \mu \sum_{k=1}^7 d^{(i,k)} \right) \rceil$$

Note in particular, that the value of this solution is reasonably close to the value of  $m^{(1)}$

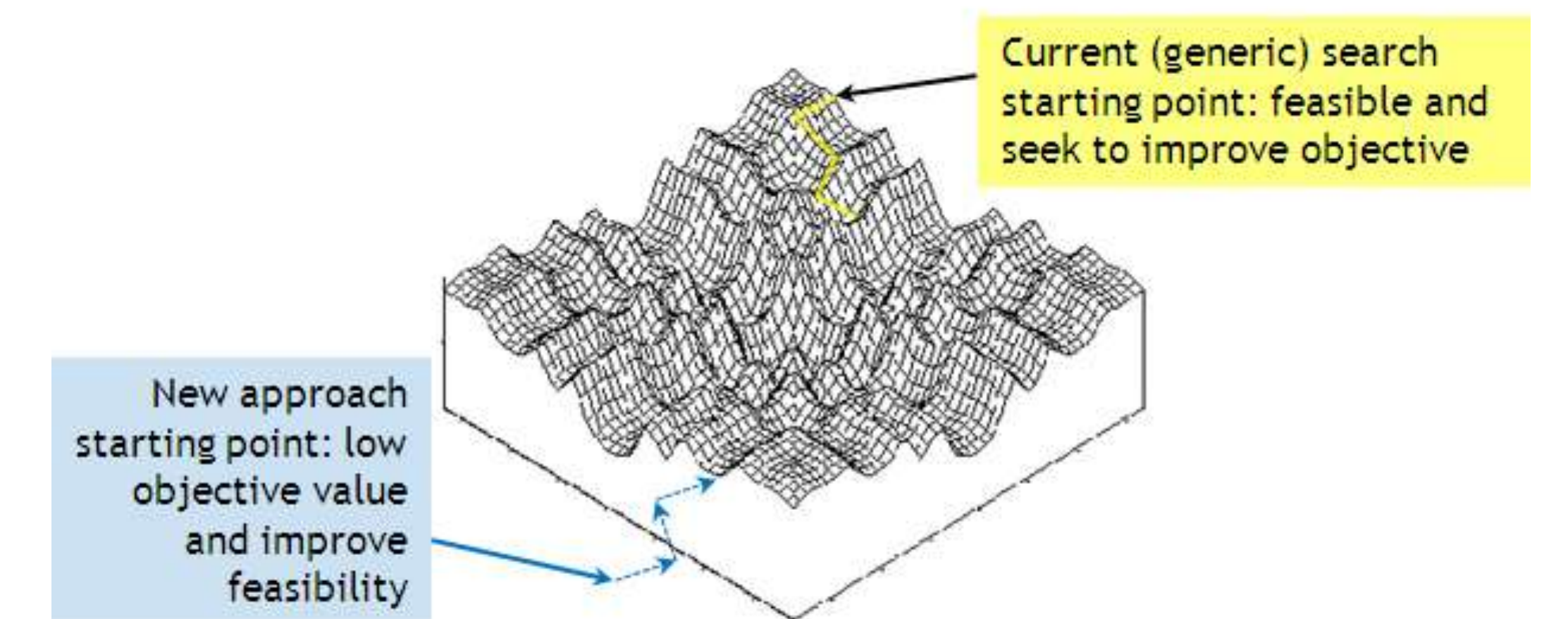
$$\sum_{i=1}^N m_i^{(2)} \leq \sum_{i=1}^N m_i^{(1)} + N - 1$$

## FUTURE WORK

•An enhanced optimization algorithm that uses the lower bounds on the pool resources per service class and per shift schedule to enhance the performance of the search of the optimal solution

•An enhanced optimization algorithm that combines the incoming tickets for all service classes to determine an optimal allocation of pool resources to take advantage of the swinging of resources to serve queues with lower skill levels

•A practical upper bound on the optimal solution for the number of service agents per skill level and shift schedule



## CONTACTS



Redha Bournas  
IBM Global Technology Services  
4205 S Miami Blvd, Durham NC 27703  
[bournas@us.ibm.com](mailto:bournas@us.ibm.com)  
919-254-1368



Brian T. Eck  
IBM Global Technology Services  
2070 Route 52, Hopewell Junction, NY 12533  
[BrianEck@us.ibm.com](mailto:BrianEck@us.ibm.com)  
845-894-8234