Contact Center Scheduling with Strict Resource Requirements

Aman Dhesi, Pranav Gupta, Amit Kumar, Gyana R Parija, Sambuddha Roy[1]

[1] High Performance Computing Group
IBM Research – India

IPCO 2011

A Typical Call Center

- This talk concerns typical scheduling scenarios encountered in call centers.

Typical Scenario in Call Centers

- We have a **forecast** of demands that will arrive at any point of time (a *timeslot*) in the future.
- A demand corresponding to a **timeslot** specifies the number of employees required in order to satisfy it.
- Any employee is required to work for shifts of a certain fixed duration (the shift-length).
- Employees may have varying constraints on their availability.
- Objective is to schedule these employees so as to *maximize* the number of timeslots for which we have the requisite number of employees.

Typical Scenario in Call Centers

- We have a **forecast** of demands that will arrive at any point of time (a *timeslot*) in the future.
- A demand corresponding to a **timeslot** specifies the number of employees required in order to satisfy it.
- Any employee is required to work for shifts of a certain fixed duration (the shift-length).
- Employees may have varying constraints on their availability.
- Objective is to schedule these employees so as to *maximize* the number of timeslots for which we have the requisite number of employees.

Typical Scenario in Call Centers

- We have a **forecast** of demands that will arrive at any point of time (a *timeslot*) in the future.
- A demand corresponding to a **timeslot** specifies the number of employees required in order to satisfy it.
- Any employee is required to work for shifts of a certain fixed duration (the shift-length).
- Employees may have varying constraints on their availability.
- Objective is to schedule these employees so as to *maximize* the number of timeslots for which we have the requisite number of employees.

Other Applications

- Sensor Networks.
- Cloud Computing
- Energy Management
- etc...

Other Applications

- Sensor Networks.
- Cloud Computing
- Energy Management
- etc...

Other Applications

- Sensor Networks.
- Cloud Computing
- Energy Management
- etc...

Roadmap

- The MaxStaff problem: definition.
- Hardness of the MaxStaff problem.
- Formulate a bicriteria version of the MaxStaff problem.
- Hardness result for the bicriteria version.
- Bicriteria Approximation Algorithms for MaxStaff
- A natural LP has unbounded integrality gap.
- Describe a configuration LP for the problem.
- Randomized Rounding on the configuration LP to obtain the desired result.

Roadmap

- The `MaxStaff` problem: definition.
- Hardness of the `MaxStaff` problem.
- Formulate a bicriteria version of the `MaxStaff` problem.
- Hardness result for the bicriteria version.
- Bicriteria Approximation Algorithms for `MaxStaff`
- A natural LP has unbounded integrality gap.
- Describe a configuration LP for the problem.
- Randomized Rounding on the configuration LP to obtain the desired result.

Roadmap

- The `MaxStaff` problem: definition.
- Hardness of the `MaxStaff` problem.
- Formulate a bicriteria version of the `MaxStaff` problem.
- Hardness result for the bicriteria version.
- Bicriteria Approximation Algorithms for `MaxStaff`
- A natural LP has unbounded integrality gap.
- Describe a configuration LP for the problem.
- Randomized Rounding on the configuration LP to obtain the desired result.

Roadmap

- The `MaxStaff` problem: definition.
- Hardness of the `MaxStaff` problem.
- Formulate a bicriteria version of the `MaxStaff` problem.
- Hardness result for the bicriteria version.
- Bicriteria Approximation Algorithms for `MaxStaff`
- A natural LP has unbounded integrality gap.
- Describe a configuration LP for the problem.
- Randomized Rounding on the configuration LP to obtain the desired result.

Roadmap

- The `MaxStaff` problem: definition.
- Hardness of the `MaxStaff` problem.
- Formulate a bicriteria version of the `MaxStaff` problem.
- Hardness result for the bicriteria version.
- Bicriteria Approximation Algorithms for `MaxStaff`
- A natural LP has unbounded integrality gap.
- Describe a configuration LP for the problem.
- Randomized Rounding on the configuration LP to obtain the desired result.

Problem Definition: MaxStaff

- $\mathcal{T}$: the set of timeslots.
- $\mathcal{R}$: the set of resources (i.e. employees).
- For each timeslot $t \in \mathcal{T}$, we are given a *demand requirement* r_t and a *profit* p_t.
- For each resource $i \in \mathcal{R}$, we are given a parameter α_i and a set of intervals $\mathcal{E}_i$, each of length L_i (the shift-length or duration).
- **Objective**: to select at most α_i intervals from $\mathcal{E}_i$ such that the number of *satisfied* timeslots is **maximized**, where a timeslot t is *satisfied* iff there are at least r_t intervals spanning it.

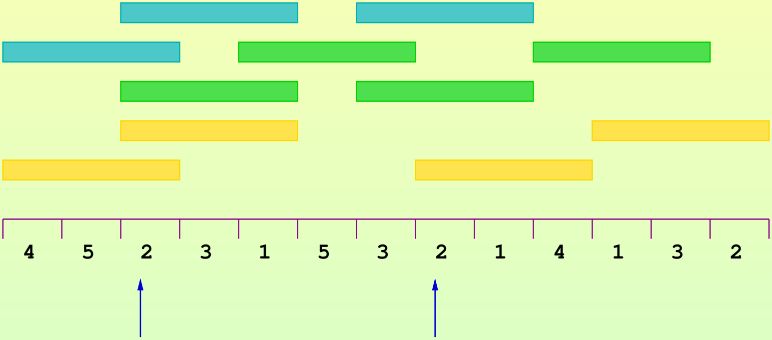
Problem Definition: `MaxStaff`

- $\mathcal{T}$: the set of timeslots.
- $\mathcal{R}$: the set of resources (i.e. employees).
- For each timeslot $t \in \mathcal{T}$, we are given a *demand requirement* r_t and a *profit* p_t.
- For each resource $i \in \mathcal{R}$, we are given a parameter α_i and a set of intervals $\mathcal{E}_i$, each of length L_i (the shift-length or duration).
- **Objective**: to select at most α_i intervals from $\mathcal{E}_i$ such that the number of *satisfied* timeslots is **maximized**, where a timeslot t is *satisfied* iff there are at least r_t intervals spanning it.

Problem Definition: `MaxStaff`

- $\mathcal{T}$: the set of timeslots.
- $\mathcal{R}$: the set of resources (i.e. employees).
- For each timeslot $t \in \mathcal{T}$, we are given a *demand requirement* r_t and a *profit* p_t.
- For each resource $i \in \mathcal{R}$, we are given a parameter α_i and a set of intervals $\mathcal{E}_i$, each of length L_i (the shift-length or duration).
- **Objective**: to select at most α_i intervals from $\mathcal{E}_i$ such that the number of *satisfied* timeslots is **maximized**, where a timeslot t is *satisfied* iff there are at least r_t intervals spanning it.

An illustration



the two timeslots cannot be satisfied simultaneously

- Here, $\alpha_i = 1$ for every $i \in \mathcal{R}$.

Hardness of `MaxStaff`

- The `MaxStaff` problem, even with $\alpha_i = 1$, $|\mathcal{E}_i| = 2$ and $L_i = 1$ (for every resource $i \in \mathcal{R}$) is NP-hard.
- Reduction from the `Maximum Independent Set` $(\mathbf{MIS})$ problem.

Bicriteria version of `MaxStaff`

- Given the NP-hardness result, we consider *bicriteria* approximation algorithms for the `MaxStaff` problem.
- Suppose OPT has to satisfy the entire coverage requirement of the timeslots; however our algorithm is allowed to satisfy only a $(1-\epsilon)$ fraction (for some ϵ) of the requirements, for any timeslot.
- Under this relaxation, can one get a constant $f(\epsilon)$ (depending on ϵ) factor approximation to OPT?
- Such an algorithm would be called an $(f(\epsilon), 1-\epsilon)$-bicriteria approximation algorithm.

Bicriteria version of `MaxStaff`: Bad news

- Even in the domain of bicriteria approximation algorithms, we extend the NP-hardness arising from the Maximum Independent Set problem to show the following:

Theorem (Lower Bound)

For any small constant $\epsilon > 0$, we cannot have a polynomial time $(\epsilon\sqrt{\log N}, 1-\epsilon)$-bicriteria approximation algorithm for the `MaxStaff` problem, unless $\mathrm{NP} \subseteq DTIME(n^{O(\log n)})$. Here N denotes the ratio $\frac{L_{max}}{L_{min}}$.

Bicriteria version of `MaxStaff`: Good news

The flavor of our result is that we show a $(\epsilon^{\log N}, 1-\epsilon)$-bicriteria approximation algorithm. We state our result for the case of **uniform shift-lengths** L (i.e. $L_i = L$ for all $i \in \mathcal{R}$):

Theorem (Upper Bound)

For any constant $\epsilon > 0$, there is a poly-time $(O(\frac{\epsilon^3}{\log\frac{1}{\epsilon}}), 1-\epsilon)$-bicriteria approximation algorithm for the `MaxStaff` problem.

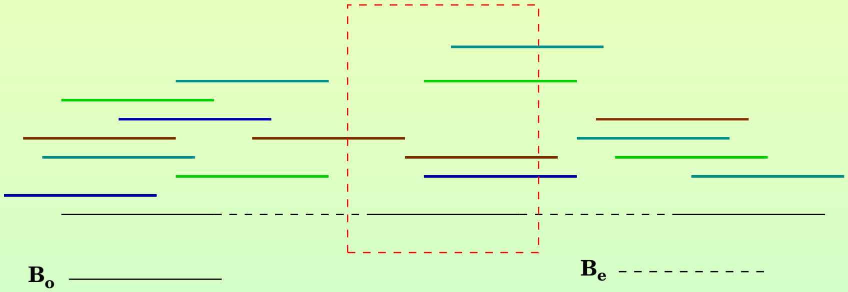
Simplifying Assumptions

- All shift-lengths are equal: $L_i = L$ for all $i \in \mathcal{R}$.

- Moreover, all demands are equal: $r_t = r$ for all $t \in \mathcal{T}$.

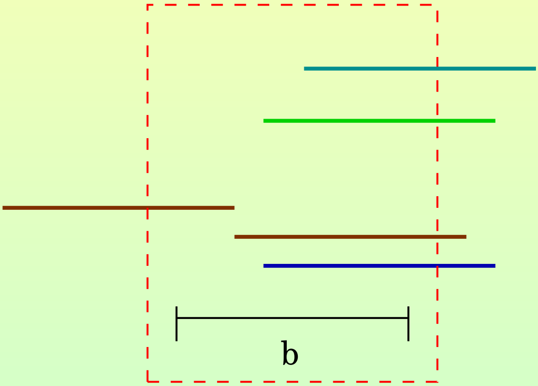
Further Simplification: Division into **blocks**

- We divide the timeline into blocks of length L each, and consider (say) only the *odd* blocks.
- At a loss of a factor of 2, we may concentrate only on solutions $\mathcal{S}$, such that no interval *contributes* to more than one block.



Situation for each **block**

- So, for any block b, the intervals active at b look as follows:

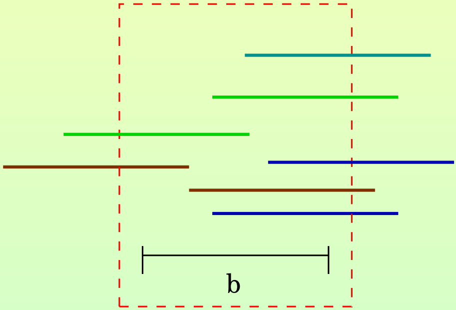


Issues with a natural LP

- A natural LP for the problem suffers from bad integrality gap – the integrality gap is as large as the shift-length L.
- The global nature of the constraints motivate us to consider a *configuration* LP.

Configuration LP: First attempt

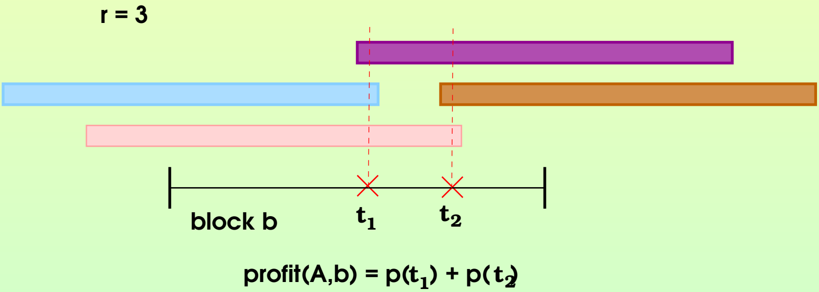
- What are **configurations**? A configuration essentially consists of a block b along with a set $\mathcal{A}$ of intervals intersecting b. Thus, variables $y(\mathcal{A}, b)$ in the LP relaxation will correspond to such tuples $(\mathcal{A}, b)$.



(A,b)

Configuration LP

- Given a set $\mathcal{A}$ of intervals intersecting b, let $\texttt{profit}(\mathcal{A}, b)$ denote the profit derived from the timeslots t in b whose demand is satisfied by the intervals in $\mathcal{A}$.
- We are now set to define the configuration LP.



The configuration LP

$$\max \sum_{b \in \mathcal{B}} \sum_{\mathcal{A}} y(\mathcal{A}, b) \cdot \texttt{profit}(\mathcal{A}, b)$$

$$\begin{array}{rc}
\forall i \in \mathcal{R} & \sum_{\mathcal{A}:\mathcal{A} \cap \mathcal{E}_i \neq \emptyset} y(\mathcal{A}, b) \leq \alpha_i \\
\forall b \in \mathcal{B} & \sum_{\mathcal{A}} y(\mathcal{A}, b) \leq 1 \\
\forall \mathcal{A}, b \in \mathcal{B} & y(\mathcal{A}, b) \in [0, 1]
\end{array}$$

- Also note that the configuration LP is a packing LP in the variables $y(\mathcal{A}, b)$; also, coefficient of each $y(\mathcal{A}, b)$ in every inequality is 1.
- Big Question: Can we solve this LP in polynomial time (even approximately, say)?

Solving the configuration LP

$$\max \sum_{b \in \mathcal{B}} \sum_{\mathcal{A}} y(\mathcal{A}, b) \cdot \texttt{profit}(\mathcal{A}, b)$$

$$\begin{array}{rl} \forall i \in \mathcal{R} & \sum_{\mathcal{A}:\mathcal{A} \cap \mathcal{E}_i \neq \emptyset} y(\mathcal{A}, b) \leq \alpha_i \\ \forall b \in \mathcal{B} & \sum_{\mathcal{A}} y(\mathcal{A}, b) \leq 1 \\ \forall \mathcal{A}, b \in \mathcal{B} & y(\mathcal{A}, b) \in [0, 1] \end{array}$$

The dual has variables z_i for every $i \in \mathcal{R}$, and variables γ_b for a block b:

$$\min \sum_{i \in \mathcal{R}} \alpha_i z_i + \sum_{b \in \mathcal{B}} \gamma_b$$

$$\begin{array}{rl} \forall \mathcal{A}, b \in \mathcal{B} & \sum_{i:\mathcal{A} \cap \mathcal{E}_i \neq \emptyset} z_i + \gamma_b \geq \texttt{profit}(\mathcal{A}, b) \\ \forall \mathcal{A}, b \in \mathcal{B} & z_i, \gamma_b \geq 0 \end{array}$$

Solving the configuration LP

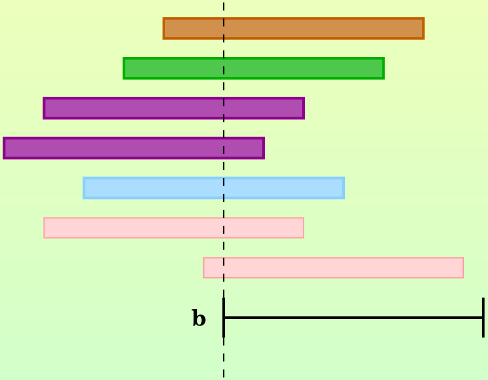
- Can we solve (the separation problem) for the dual LP? The essential problem is:
- We are given values z_i's and γ_b's as inputs, and we have to decide if there is a violation of the inequality $\sum_{i:\mathcal{A}\cap\mathcal{E}_i\neq\emptyset} z_i + \gamma_b \geq \texttt{profit}(\mathcal{A}, b)$, for some configuration $(\mathcal{A}, b)$.
- I.e. we are searching for a collection $\mathcal{A}$ of intervals such that the following holds:

$$\sum_{i:\mathcal{A}\cap\mathcal{E}_i\neq\emptyset} z_i + \gamma_b < \texttt{profit}(\mathcal{A}, b)$$

- (Think of the z_i's as costs, and we are looking for a collection of intervals that gives us “progress" – the **cost** of the intervals is *less* than the **profit** derived from that collection.)

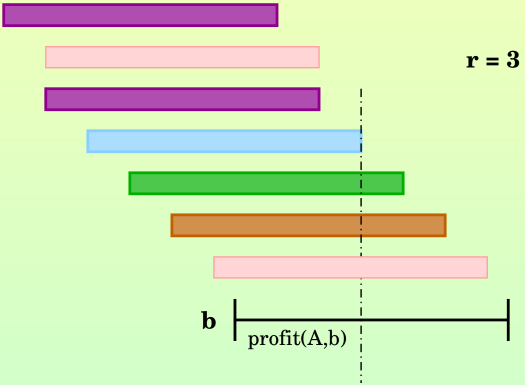
Configuration LP: a fictitious case

- Let us consider a (fictitious) special case: where for every block b, all the intervals in the universe (each of length L, the same as the size of the block) intersect the left endpoint of the block b.



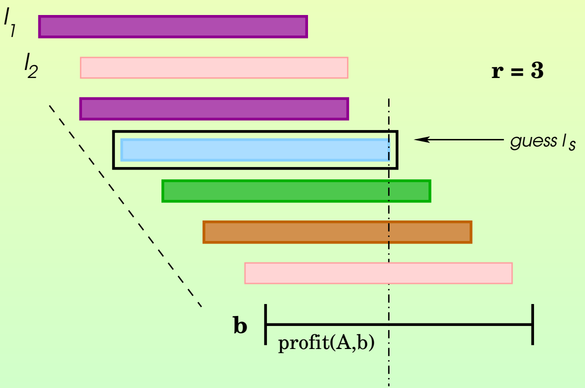
Solving the configuration LP

- Arrange all the intervals (call this collection $\mathcal{S}$) intersecting b in increasing order of their endpoints. In order to derive non-zero $\texttt{profit}(\mathcal{A}, b)$, $\mathcal{A}$ should consist of r intervals from $\mathcal{S}$.



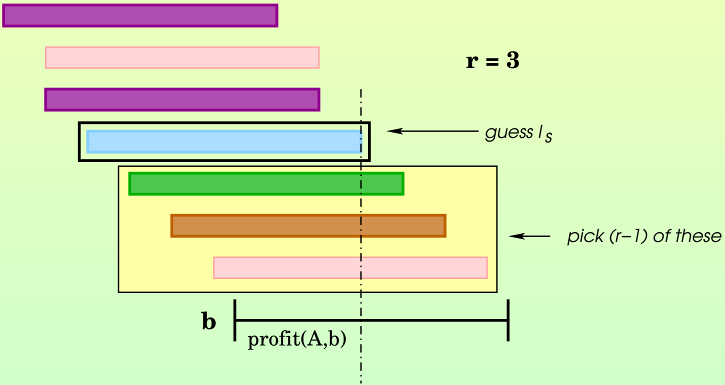
Solving the configuration LP

- Guess the block b that gives rise to a violating collection $\mathcal{A}$.
- Let the ordering of the intervals from left to right be $I_1, I_2, \cdots, I_\ell$. Let I_s be the interval of least index s in a potential violating collection $\mathcal{A}$.



Solving the configuration LP

- Among $I_{s+1}, \cdots, I_\ell$, we have to pick $(r-1)$ intervals so as to cover some part of b.
- The part covered of b is precisely the intersection of b with I_s (this gives us the profit, $\texttt{profit}(\mathcal{A}, b)$).

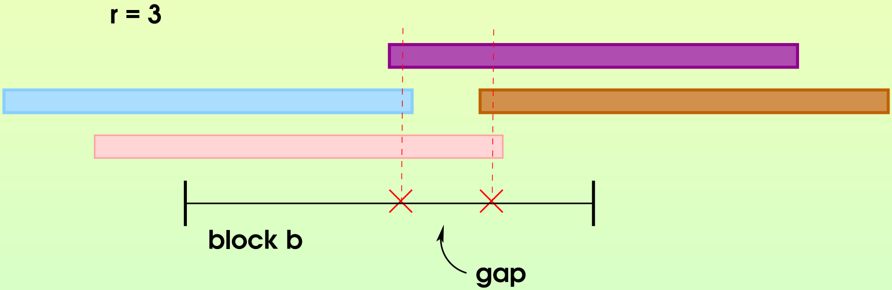


Solving the configuration LP

- Choose (r-1) intervals among $I_{s+1}, \cdots, I_\ell$ with the lowest values of z_I. This, along with I_s, constitutes the potentially violating collection of intervals $\mathcal{A}$.
- Check if the inequality $\sum_{i:\mathcal{A}\cap\mathcal{E}_i\neq\emptyset} z_i + \gamma_b \geq \texttt{profit}(\mathcal{A}, b)$ holds, given the value of γ_b.

However. . .

- In reality, intervals could intersect a block b from either side.
- This complicates matters. . .



However. . .

- It is not easy to separate the dual LP any more.
- Thereby, we reformulate a new configuration LP, with a *proxy* objective function.
- (Also, note that “bicriteria" has not figured so far in our considerations. . .)

The new objective function

The essential reason why for the fictitious case, we could separate the dual LP, was that we were guaranteed two things:

- the timeslots *realising* $\texttt{profit}(\mathcal{A}, b)$ were **contiguous**, thus forming a (sub)-interval of $\mathcal{T}$, and
- each resource interval i contributing to covering these timeslots covered the *entire* sub-interval.

We would like to emulate a similar scenario for when the resource intervals can intersect a block b from either side.

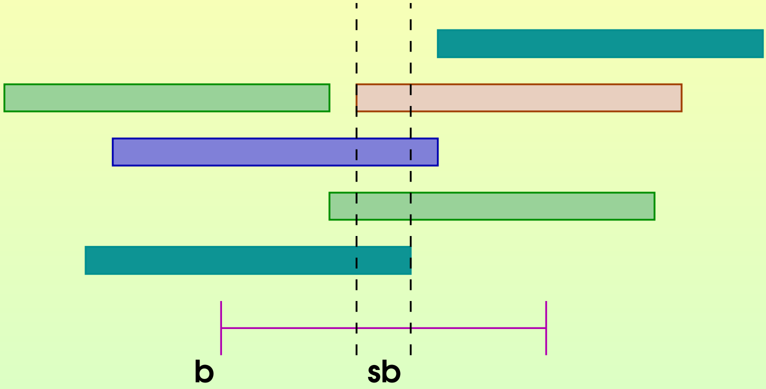
The new objective function

We define a modified profit function $p(\mathcal{A}, b)$ as follows:

Definition: $p(\mathcal{S}, b)$

Given a collection of intervals $\mathcal{S}$, a block b and a sub-block sb of b, let $\mathcal{S}'(sb) \subseteq \mathcal{S}$ denote the collection of intervals that *entirely* span the sub-block sb.
Let $\texttt{profit}^{\epsilon}(\mathcal{S}, sb)$ denote the total profit of timeslots in sb if $|\mathcal{S}'(sb)| \geq (1 - \epsilon)r$, or else 0.
Let $p(\mathcal{S}, b)$ denote the maximum of $\texttt{profit}^{\epsilon}(\mathcal{S}, sb)$, over *all possible sub-blocks sb*.

The new objective function



We claim:

Lemma

$$p(\mathcal{S}, b) \geq \epsilon \cdot \mathtt{profit}(\mathcal{S}, b)$$

The new objective function

Given the above, the new objective function reads as follows:

$$\max \sum_{b \in \mathcal{B}} \sum_{\mathcal{A}} y(\mathcal{A}, b) \cdot p(\mathcal{A}, b)$$

along with the earlier constraints:

$$\begin{array}{rc} \forall i \in \mathcal{R} & \sum_{\mathcal{A}:\mathcal{A} \cap \mathcal{E}_i \neq \emptyset} y(\mathcal{A}, b) \leq \alpha_i \\ \forall b \in \mathcal{B} & \sum_{\mathcal{A}} y(\mathcal{A}, b) \leq 1 \\ \forall \mathcal{A}, b \in \mathcal{B} & y(\mathcal{A}, b) \in [0, 1] \end{array}$$

The new objective function

Given the above, the new objective function reads as follows:

$$\max \sum_{b \in \mathcal{B}} \sum_{\mathcal{A}} y(\mathcal{A}, b) \cdot p(\mathcal{A}, b)$$

along with the earlier constraints:

$$\begin{array}{rc} \forall i \in \mathcal{R} & \sum_{\mathcal{A}:\mathcal{A} \cap \mathcal{E}_i \neq \emptyset} y(\mathcal{A}, b) \leq \alpha_i \\ \forall b \in \mathcal{B} & \sum_{\mathcal{A}} y(\mathcal{A}, b) \leq 1 \\ \forall \mathcal{A}, b \in \mathcal{B} & y(\mathcal{A}, b) \in [0, 1] \end{array}$$

Solving the modified configuration LP

- Given the modified objective function, the earlier considerations now hold, and we can solve the dual LP in polynomial time.
- Of course, thereby, we lose a factor of ϵ in our original objective function.
- Thus we get the optimal fractional solution to the modified configuration LP.

Randomized Rounding for the new LP

Given the modified configuration LP, we apply (standard) randomized rounding to thereby obtain integral solutions (with corresponding loss in quality) from the fractional solutions $y(\mathcal{A}, b)$ provided by the LP solution.
Thus we get the following result:

Theorem: case of uniform demands

For any constant $\epsilon > 0$, there is an $(O(\epsilon), 1 - \epsilon)$-bicriteria approximation algorithm for the `MaxStaff` problem.

Extensions

- Arbitrary demands: We lose an extra ϵ^2 factor here.
- Arbitrary shift-lengths: The approximation factor becomes worse by $N = \frac{L_{\max}}{L_{\min}}$.
- “Heights" of the resources: In the above, we considered the resources to have uniform height 1. However, in reality, resources/employees have different proficiencies, and these correspond to the resources having “heights". We can extend our algorithms for height 1 to this case.
- In reality, the different shifts of a single resource are stipulated to have enough “gap" (say Δ) between them. Our algorithms extend to this case too.

THANK YOU

A natural LP

We describe a natural LP relaxation for the problem. The LP has a variable x_t for each timeslot t, and a variable $y(I)$ for every interval I.

$$\max x_t p_t$$

$$\begin{array}{rc}
\forall t \in \mathcal{T} & \sum_{i \in \mathcal{R}} \min\{x_t, \sum_{I \in \mathcal{E}_i, t \in I} y(I)\} \geq x_t r_t \\
\forall i \in \mathcal{R} & \sum_{I \in \mathcal{E}_i} y(I) \leq \alpha_i \\
\forall t \in \mathcal{T}, I & x_t, y(I) \in [0, 1]
\end{array}$$

Note that intervals in $\mathcal{E}_i$ do not need to be disjoint.

The case of arbitrary demands

- In the case of arbitrary demands, one possible approach is to scale-and-group the demands, and for each demand “level" solve the configuration LP and take the union of all the solutions.
- However this may not work because of the constraint that any resource *i* may be picked at most α_i times.
- Thus, we consider a single configuration LP that has a slightly modified notion of “configuration", and then we (a) solve the LP as before, and (b) perform randomized rounding.