

Задача размещения производства

Дано: $I = \{1, \dots, m\}$ — множество районов (городов, областей), где можно открыть производство некоторой продукции;

$J = \{1, \dots, n\}$ — множество потребителей этой продукции (клиентов);

$f_i \geq 0$ — затраты на организацию производства в пункте i .

$c_{ij} \geq 0$ — производственно–транспортные расходы на удовлетворение спроса j -го клиента i -м предприятием.

$p > 0$ — максимально возможное число предприятий.



Найти: подмножество предприятий $S \subset I$, $|S| \leq p$, которое позволяет удовлетворить спрос всех клиентов с минимальными суммарными затратами.

Переменные задачи:

$$x_i = \begin{cases} 1, & \text{если открываем предприятие } i, \\ 0 & \text{в противном случае} \end{cases}$$

$$z_{ij} = \begin{cases} 1, & \text{если предприятие } i \text{ обслуживает клиента } j, \\ 0 & \text{в противном случае} \end{cases}$$

Математическая модель

$$\min \left\{ \sum_{i \in I} \sum_{j \in J} c_{ij} z_{ij} + \sum_{i \in I} f_i x_i \right\}$$

при ограничениях:

$$\sum_{i \in I} z_{ij} = 1, \quad j \in J;$$

$$x_i \geq z_{ij}, \quad i \in I, \quad j \in J;$$

$$\sum_{i \in I} x_i \leq p;$$

$$z_{ij}, x_i \in \{0, 1\}, \quad i \in I, j \in J.$$

Упражнение. Замена $z_{ij} \in \{0, 1\}$ на $z_{ij} \geq 0$ не меняет минимума суммарных затрат.

Задача размещения с предпочтениями клиентов

Дано: $I = \{1, \dots, m\}$ — множество мест, где можно открыть производство некоторой продукции;
 $J = \{1, \dots, n\}$ — множество потребителей этой продукции (клиентов);
 $f_i \geq 0$ — затраты на организацию производства в пункте i .
 $c_{ij} \geq 0$ — производственно–транспортные расходы на удовлетворение спроса j -го клиента i -м предприятием.
 $p > 0$ — максимально возможное число предприятий.



$d_{ij} \geq 0$ — предпочтения j -го клиента на множестве предприятий:

$\min_{i \in I} d_{ij}$ — наиболее желаемый поставщик;

$\max_{i \in I} d_{ij}$ — наименее желаемый поставщик.

Найти: Подмножество $S \subset I$, $|S| \leq p$, открываемых предприятий, которые позволили бы удовлетворить спрос всех клиентов с минимальными суммарными затратами.

Переменные задачи:

$$x_i = \begin{cases} 1, & \text{если открываем предприятие } i, \\ 0 & \text{в противном случае} \end{cases}$$

$$z_{ij} = \begin{cases} 1, & \text{если предприятие } i \text{ обслуживает клиента } j, \\ 0 & \text{в противном случае} \end{cases}$$

Математическая модель

$$\min \left\{ \sum_{i \in I} \sum_{j \in J} c_{ij} z_{ij}^* + \sum_{i \in I} f_i x_i \right\}$$

при ограничениях:

$$\sum_{i \in I} x_i \leq p;$$

$$x_i \in \{0,1\}, \quad i \in I,$$

где z_{ij}^* — оптимальное решение задачи потребителя:

$$\min \sum_{j \in J} \sum_{i \in I} d_{ij} z_{ij}$$

при ограничениях:

$$\sum_{i \in I} z_{ij} = 1, \quad j \in J;$$

$$x_i \geq z_{ij}, \quad i \in I, \quad j \in J;$$

$$z_{ij} \in \{0,1\}, \quad i \in I, \quad j \in J.$$

Генетический алгоритм для задач размещения

Идея заимствована у живой природы и состоит в организации эволюции, целью которой является получение оптимального решения в сложной комбинаторной задаче:

$$\min \{f(S), S \in Sol\}.$$

Начальная популяция $P = \{S_1, S_2, \dots, S_k\}$ — набор допустимых решений исходной задачи.

Шаг эволюции: выбираем из популяции два решения, скрещиваем их, применяем мутацию, локальную перестройку и добавляем в популяцию, затем наихудшее решение удаляем из популяции.



Общая схема алгоритма

1. Выбрать начальную популяцию P и запомнить рекорд $F^* = \min_{i=1, \dots, k} f(S_i)$.
2. Пока не выполнен критерий останова делать следующее:
 - 2.1. Выбрать “родителей” S_{i_1}, S_{i_2} из популяции.
 - 2.2. Применить к S_{i_1}, S_{i_2} оператор скрещивания и получить новое решение S' .
 - 2.3. Применить к S' оператор мутации и получить новое решение S'' .
 - 2.4. Применить к S'' оператор локального улучшения и получить новое решение S''' .
 - 2.5. Если $f(S''') < F^*$, то сменить рекорд $F^* := f(S''')$.
 - 2.6. Добавить S''' к популяции и удалить из нее наихудшее решение.

Оператор скрещивания

Пусть S_1, S_2 — два решения, задаваемые векторами $X^1, X^2 \in \{0,1\}^n$.

Одноточечный оператор скрещивания: выбираем случайным образом координату $1 \leq l \leq n$ и новый вектор X' получает первые l координат от вектора X^1 , а остальные от вектора X^2 .

$$\begin{array}{l} X^1: (0 \ 1 \ 0 \ 0 \mid 1 \ . \ . \ . \ 0 \ 1 \ 1) \\ X^2: (1 \ 1 \ 0 \ 1 \mid 0 \ . \ . \ . \ 1 \ 1 \ 1) \\ X': (0 \ 1 \ 0 \ 0 \mid 0 \ . \ . \ . \ 1 \ 1 \ 1) \end{array}$$

$\underbrace{\hspace{10em}}_l$

Аналогично определяется двухточечный, трехточечный и т.д. операторы.

Равномерный оператор скрещивания: новое решение X' в каждой координате получает с вероятностью 0.5 значение одного из родителей.

Выбор родителей

Турнирная селекция: из популяции P случайным образом выбирается некоторое подмножество $P' \subseteq P$ и родителем назначается наилучшее решение в P' :

$$S_i = \min_{S \in P'} f(S).$$

Пропорциональная селекция: из популяции P случайным образом выбираются два родителя. Для решения S_i вероятность быть выбранным обратно пропорциональна значению целевой функции $f(S_i)$.

Варианты: Лучший в популяции + случайно выбранный.

Случайно выбранный + наиболее удаленный от него и др.

Оператор мутации

Вероятностный оператор мутации случайным образом вносит изменения в допустимое решение задачи. Например, с малой вероятностью $q < \frac{1}{n}$ в каждой координате значение $X_i \in \{0,1\}$ заменяется на противоположное $1 - X_i$. Если в решении требуется сохранить $\sum_{i \in I} x_i = p$, то случайным образом выбирается координата i_1 такая, что $X_{i_1} = 1$ и координата i_2 такая, что $X_{i_2} = 0$ и производится замена $X_{i_1} := 0, X_{i_2} := 1$.

$$\begin{array}{l} X' = (0 \ 1 \ 0 \ 1 \ 0 \ \dots \ 0 \ 1 \ 1) \\ X'' = (1 \ 1 \ 0 \ 0 \ 0 \ \dots \ 0 \ 1 \ 1) \\ \qquad \qquad \qquad \underbrace{\qquad \qquad \qquad}_{i_1 \qquad \qquad i_2} \end{array}$$

Локальное улучшение

Для решения S обозначим через $N(S)$ его окрестность, например, множество всех решений S' , находящихся от S на расстоянии не более 2 (3, 4, 5...)

Алгоритм локального спуска

1. Положить $S := S''$;
2. Найти в окрестности решения S наилучшего соседа \bar{S}

$$f(\bar{S}) = \min \{f(\tilde{S}), \tilde{S} \in N(S)\};$$

3. Если $f(\bar{S}) < f(S)$, то положить $S := \bar{S}$ и вернуться на шаг 2, иначе STOP, получен локальный минимум.

Вопросы

- Оцените трудоемкость шагов 2.1, 2.2, 2.3, 2.4.
- Является ли генетический алгоритм полиномиальным?
(*Да или Нет?*)
- Является ли генетический алгоритм точным? (*Да или Нет?*)
- Задача поиска наилучшего потомка S'' для выбранных родителей является NP-трудной задачей? (*Да или Нет?*)
- Можно ли этим алгоритмом решать задачу коммивояжера ? (*Да или Нет?*)

Задача размещения в условиях конкуренции

$I = \{1, \dots, m\}$ — множество пунктов размещения;

$J = \{1, \dots, n\}$ — множество клиентов;

c_{ij} — расстояние от пункта i до клиента j ;

p — число предприятий, открываемых ЛПР₁;

r — число предприятий, открываемых ЛПР₂;

d_j — доход от обслуживания клиента j ;

Сначала ЛПР₁ принимает решение об открытии своих предприятий. Затем, зная это решение, ЛПР₂ открывает свои предприятия так, чтобы получить максимальный доход. Клиент знает оба решения и выбирает ближайшее к себе предприятие. Задача состоит в том, чтобы найти решение для ЛПР₁ с максимальным доходом.



Переменные задачи

$$x_i = \begin{cases} 1, & \text{если ЛПР}_1 \text{ открыл предприятие в пункте } i \\ 0 & \text{в противном случае} \end{cases}$$

$$y_i = \begin{cases} 1, & \text{если ЛПР}_2 \text{ открыл предприятие в пункте } i \\ 0 & \text{в противном случае} \end{cases}$$

$$z_j = \begin{cases} 1, & \text{если клиент } j \text{ обслуживается ЛПР}_1 \\ 0, & \text{если клиент } j \text{ обслуживается ЛПР}_2 \end{cases}$$

Для вектора x положим

$I_j(x) = \{i \in I \mid c_{ij} < \min(c_{kj} \mid x_k = 1)\}$ — множество пунктов размещения, которые находятся ближе к клиенту j , чем ближайшее открытое предприятие ЛПР₁.

Математическая модель двухуровневого программирования

$$\max_x \sum_{j \in J} d_j z_j^*(x)$$

при ограничениях
$$\sum_{i \in I} x_i = p, \quad x_i \in \{0,1\}, \quad i \in I$$

где z_j^* — оптимальное решение задачи ЛПР₂:

$$\max_{z,y} \sum_{j \in J} d_j (1 - z_j)$$

при ограничениях
$$1 - z_j \leq \sum_{i \in I_j(x)} y_i, \quad j \in J$$

$$\sum_{i \in I} y_i = r$$

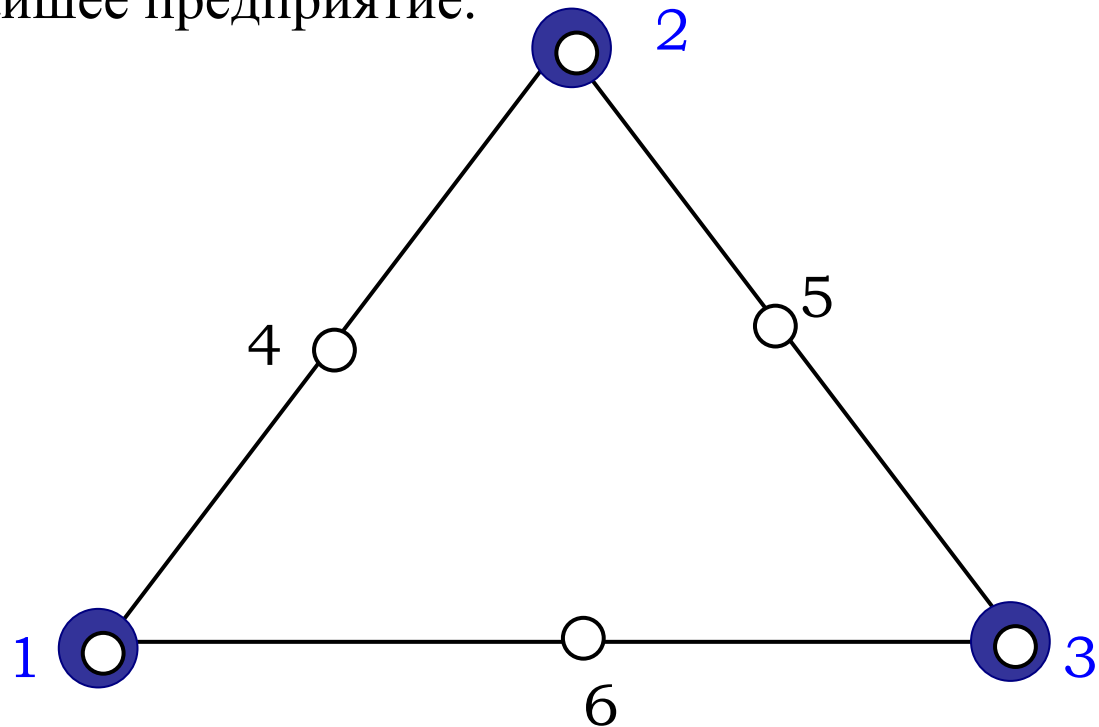
$$x_i + y_i \leq 1, \quad i \in I, \quad y_i, z_j \in \{0,1\}$$

«Безнадежный» пример

$I = \{1, 2, 3, 4, 5, 6\}$ — места размещения предприятий;

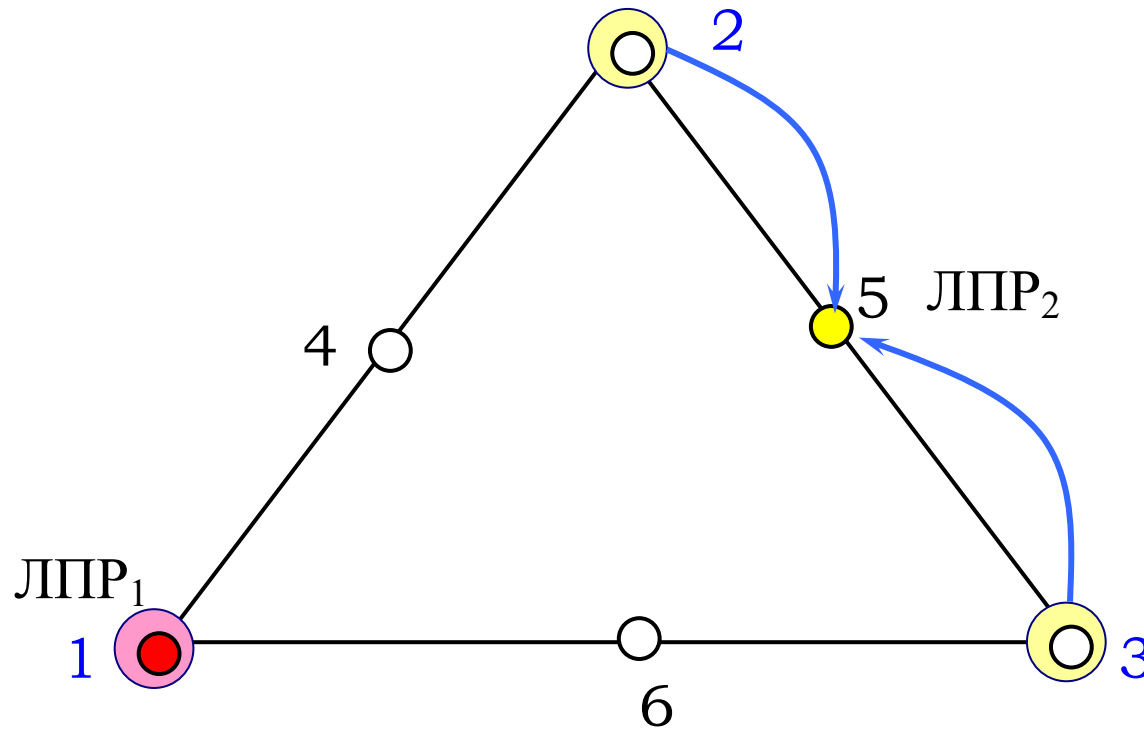
$J = \{1, 2, 3\}$ — клиенты.

Клиенты выбирают ближайшее предприятие.

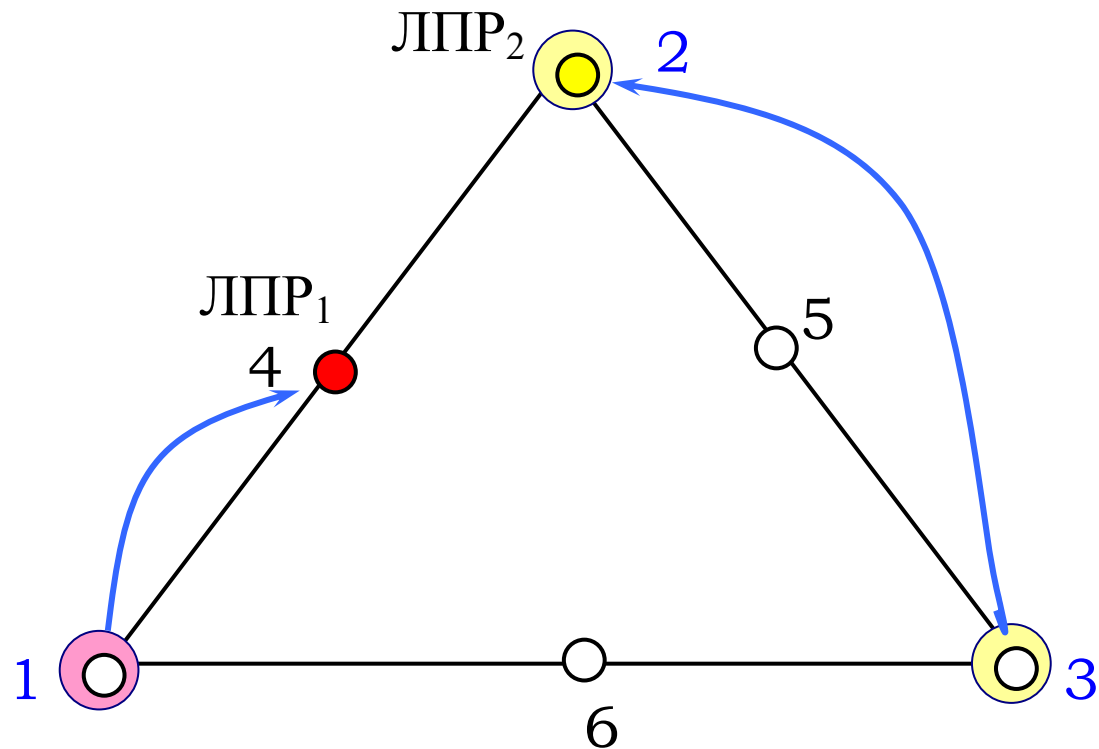


При $p = r = 1$ ЛПР₁ всегда проигрывает!

Если ЛПР₁ ставит предприятие в вершине треугольника, то ЛПР₂ ставит свое предприятие на противоположной стороне треугольника и захватывает двух клиентов, в то время как ЛПР₁ получает только одного.



Если ЛПР₁ ставит предприятие на стороне треугольника, то ЛПР₂ ставит свое предприятие в соседней вершине треугольника и тоже захватывает двух клиентов, в то время как ЛПР₁ получает только одного



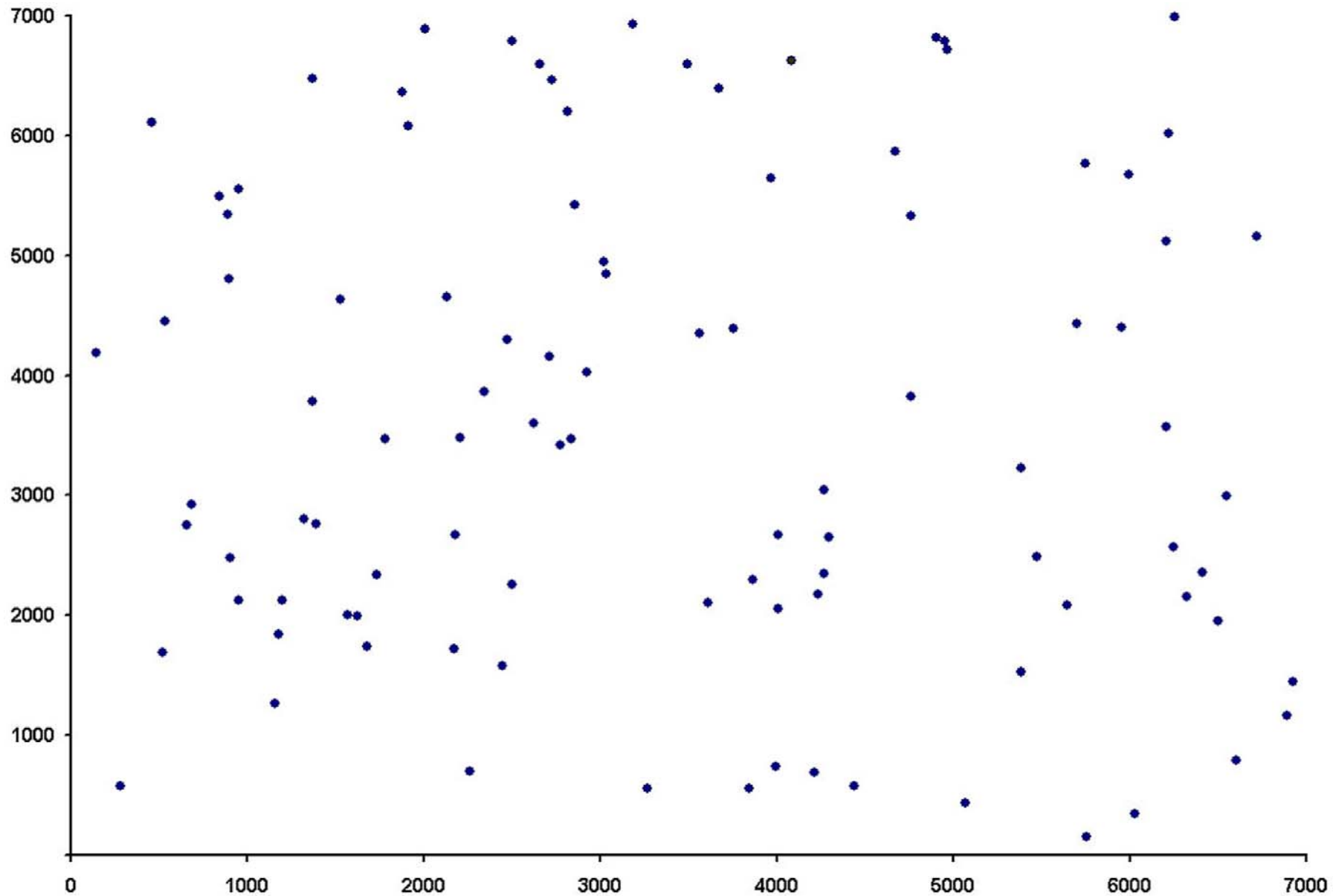
Численные методы

При заданном решении x для ЛПР₁ задача ЛПР₂ является NP-трудной задачей целочисленного линейного программирования.

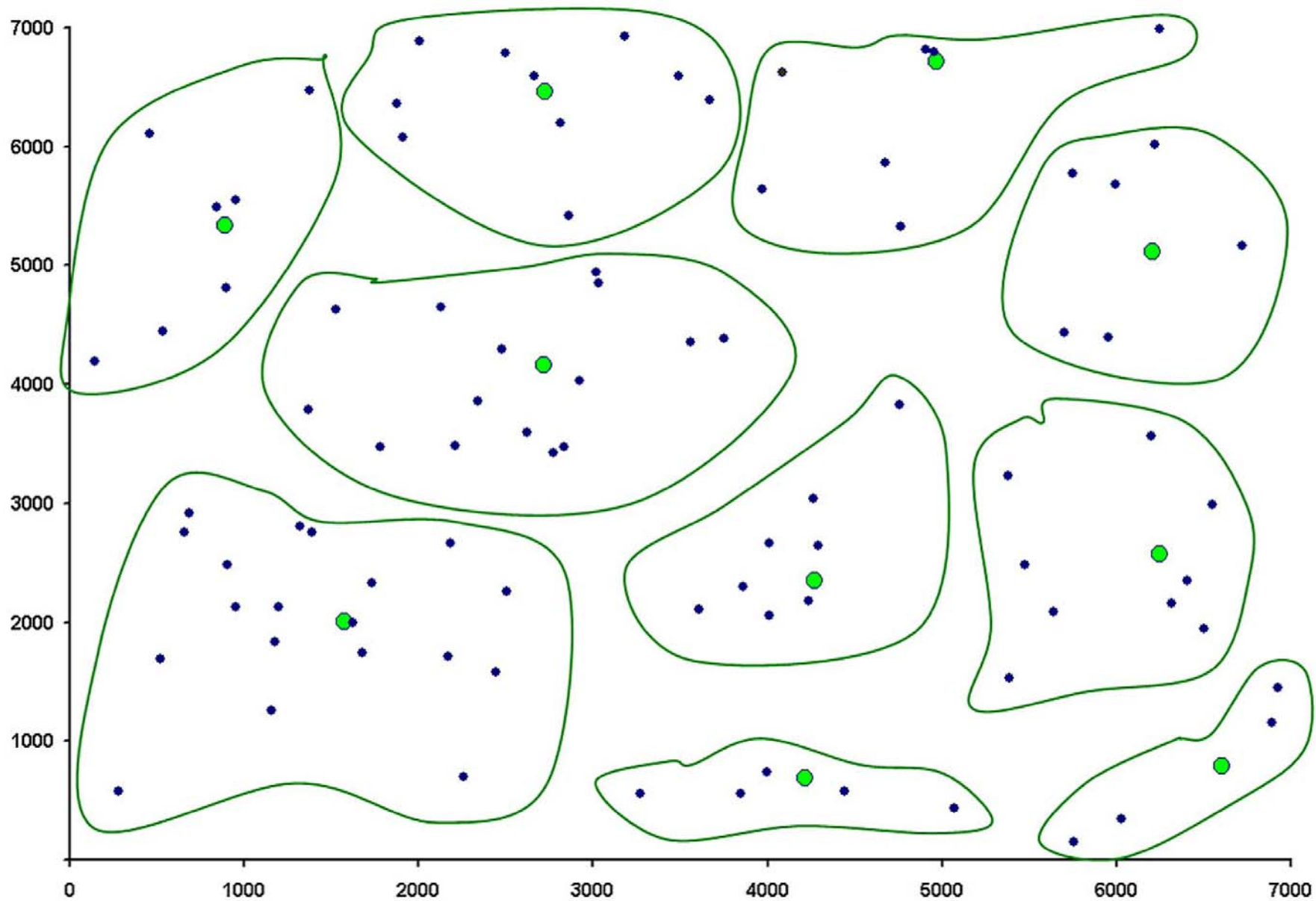
Зная ее оптимальное решение, можно вычислить доход ЛПР₂ и ЛПР₁.

- Генетический локальный поиск в пространстве переменных x для ЛПР₁.
- Имитация отжига в пространстве переменных x для ЛПР₁.
- Поиск с чередующимися окрестностями в пространстве переменных x для ЛПР₁.

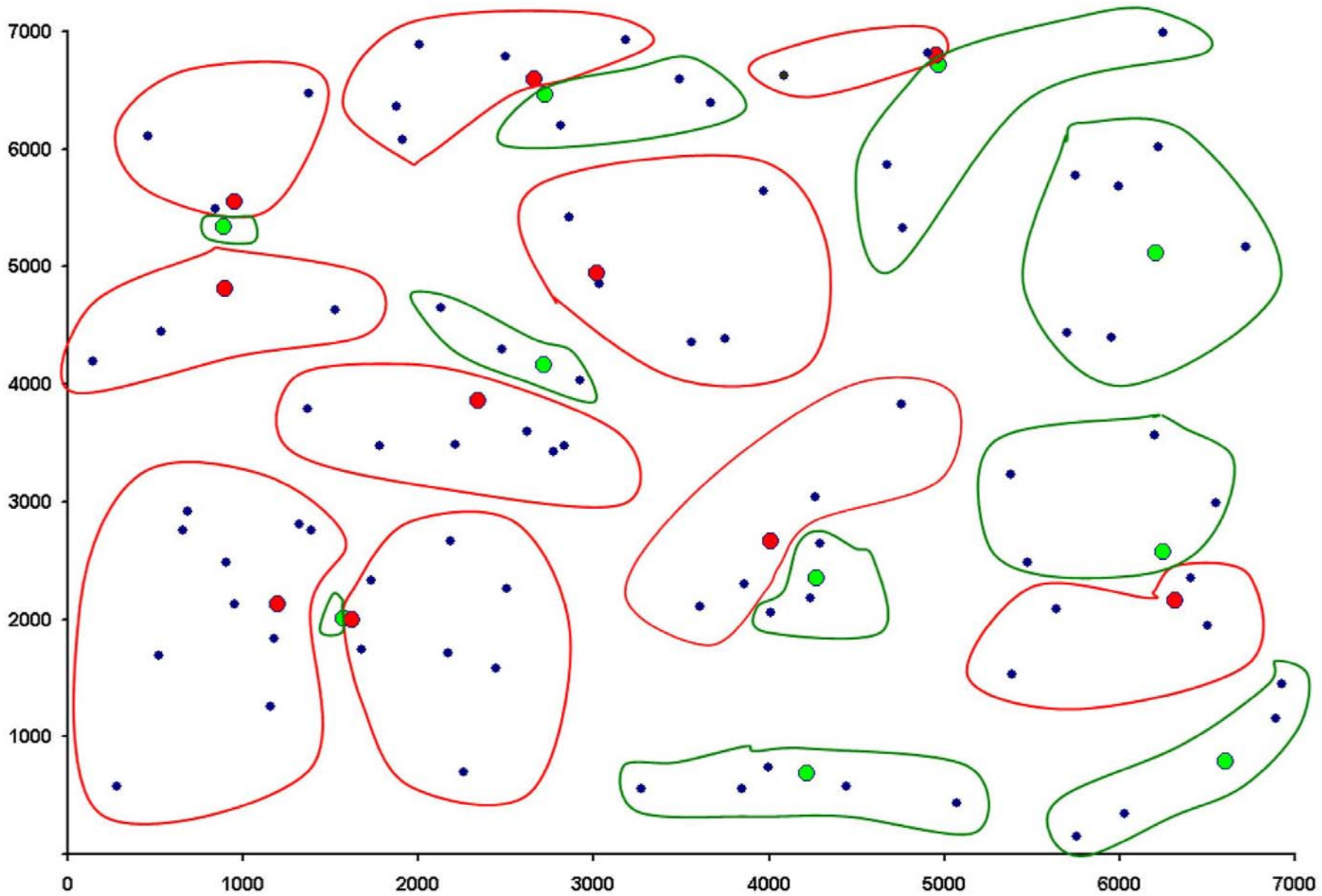
Euclidean instance, $|I| = 100$, $|J| = 100$



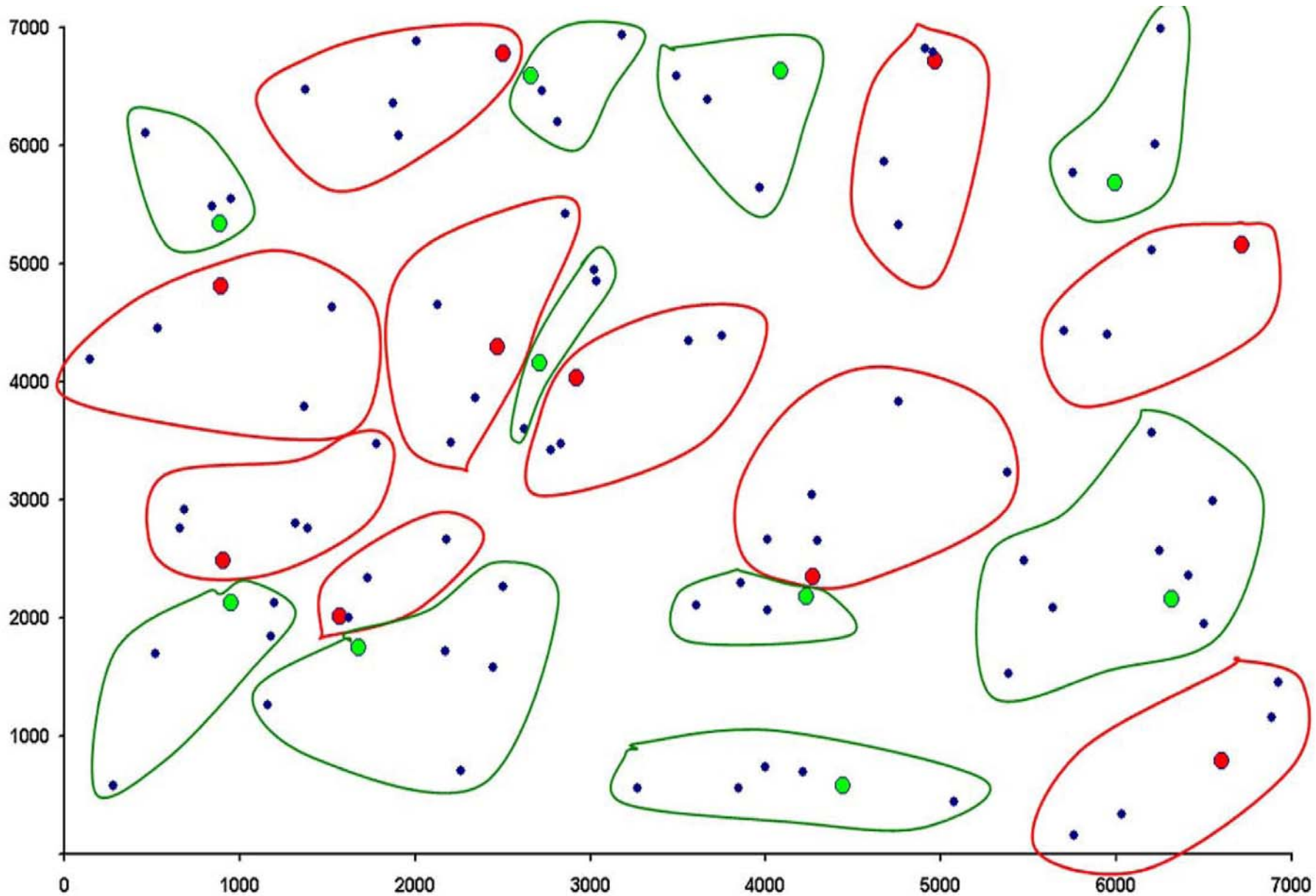
The leader ignores the follower



Optimal solution of the follower. Market share of the leader is 41 %



Optimal solution of the leader. Market share of the leader is 50 %



Точный метод

Пусть \mathcal{F} — непустое семейство решений ЛПР₂.

Для $y \in \mathcal{F}$ положим

$$I_j(y) = \{i \in I \mid c_{ij} \leq \min_{l \in I} c_{lj} \mid y_l = 1\}$$

Это множество предприятий, позволяющих ЛПР₁ удержать клиента j , если ЛПР₂ использует решение y .

Дополнительные переменные:

D — доход ЛПР₁

$$z_{ij} = \begin{cases} 1, & \text{если предприятие } i \text{ ЛПР}_1 \text{ ближайшее для клиента } j \\ 0 & \text{в противном случае} \end{cases}$$

Переформулировка задачи:

$$\max D \tag{1}$$

при ограничениях $\sum_{j \in J} \sum_{i \in I_j(y)} d_j z_{ij} \geq D, \quad \forall y \in \mathcal{F},$ (2)

$$\sum_{i \in I} z_{ij} = 1, \quad j \in J, \tag{3}$$

$$x_i \geq z_{ij}, \quad i \in I, j \in J, \tag{4}$$

$$\sum_{i \in I} x_i = p. \tag{5}$$

Если \mathcal{F} — все решения ЛПР₂, то получаем эквивалентную переформулировку.

Пусть \mathcal{F} — некоторое семейство решений ЛПР₂.

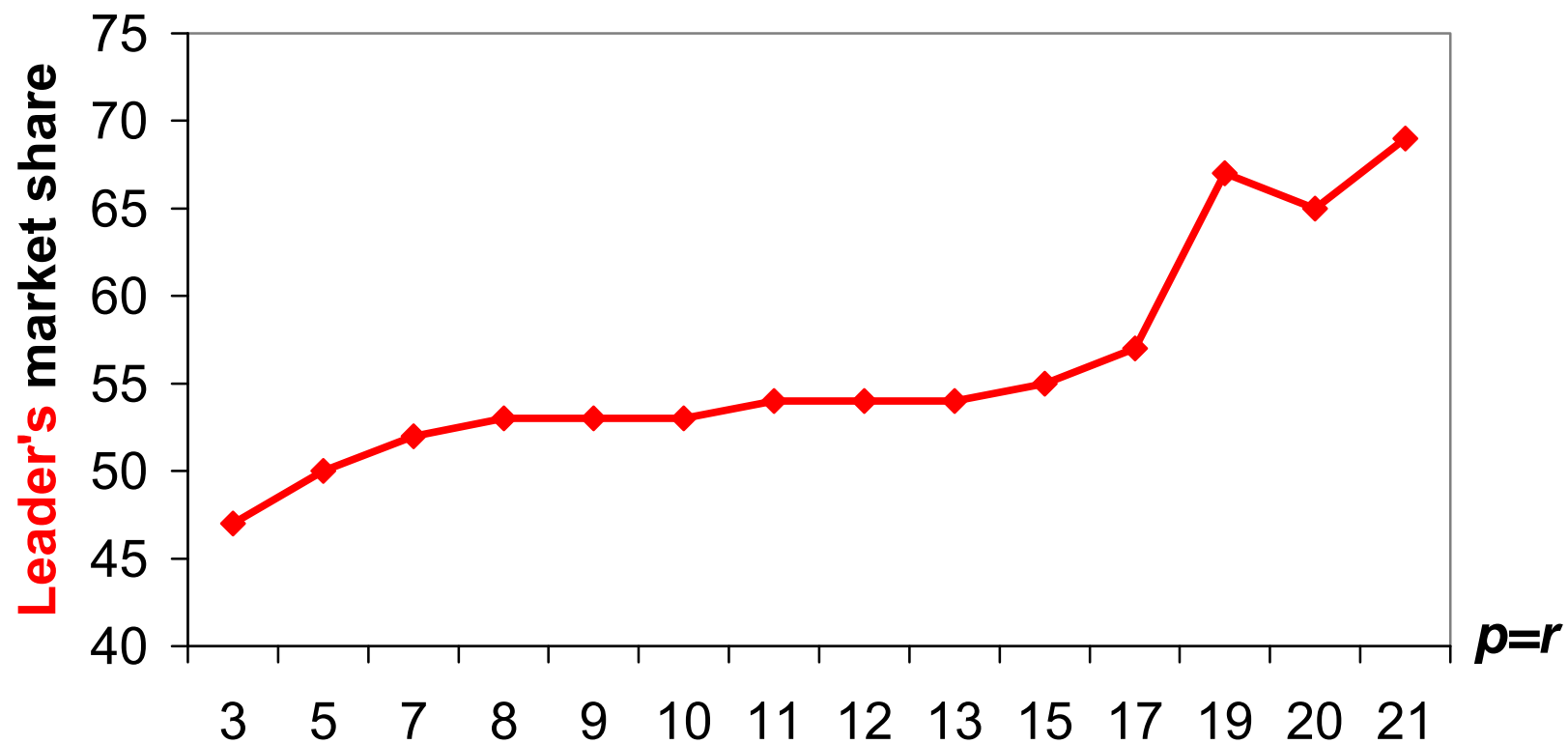
Тогда $D(\mathcal{F})$ — верхняя оценка максимального дохода ЛПР₁.

Пусть $x(\mathcal{F})$ — оптимальное решение для заданного \mathcal{F} и $D'(x(\mathcal{F}))$ — доход ЛПР₁ на решении $x(\mathcal{F})$. $D'(x(\mathcal{F}))$ — нижняя оценка оптимума для ЛПР₁.

Итерационный метод:

0. Выбрать начальное семейство \mathcal{F} и положить $D^* := 0$
1. Решить задачу (1)–(5), найти $D(\mathcal{F})$ и $x(\mathcal{F})$
2. Решить задачу ЛПР₂, вычислить $y(x(\mathcal{F}))$ и $D'(x(\mathcal{F}))$
3. Если $D^* < D'(x(\mathcal{F}))$, то $D^* := D'(x(\mathcal{F}))$
4. Если $D(\mathcal{F}) = D^*$, то STOP
5. Добавить $y(x(\mathcal{F}))$ в семейство \mathcal{F} и вернуться на шаг 1.

Доля рынка ЛПР₁



Вопросы

- Задача размещения в условиях конкуренции может быть решена полным перебором вариантов, и их число не превышает $C_n^p C_{n-p}^r$
(Да или Нет?)
- Задано число K и решение x для ЛПР₁. Рассмотрим следующую задачу распознавания: правда ли, что ЛПР₂ может получить долю рынка не менее K ? Эта задача распознавания принадлежит классу NP
(Да или Нет?)
- Задано число K . Рассмотрим следующую задачу распознавания: правда ли, что ЛПР₁ может получить долю рынка не менее K ? Эта задача распознавания принадлежит классу NP (Да или Нет?)

Квадратичная задача о назначениях

Дано: $I = \{1, \dots, n\}$ — множество зданий, где могут размещаться цеха;

$J = \{1, \dots, n\}$ — множество производственных цехов;

a_{kl} — расстояние между зданиями $k, l \in I$;

b_{ij} — объем продукции, транспортируемый из цеха i в цех j , где $i, j \in J$.



Найти: Размещение цехов по зданиям, при котором суммарный объем перевозок продукции был бы минимальным.

Переменные задачи:

$$x_{ik} = \begin{cases} 1, & \text{если цех } i \text{ размещается в здании } k, \\ 0 & \text{в противном случае} \end{cases}$$

Математическая модель

$$\min \sum_{i \in J} \sum_{j \in J} \sum_{l \in I} \sum_{k \in I} a_{kl} b_{ij} x_{ik} x_{jl}$$

при ограничениях:

$$\sum_{i \in J} x_{ik} = 1, \text{ для всех } k \in I;$$

$$\sum_{k \in I} x_{ik} = 1, \text{ для всех } i \in J;$$

$$x_{ik} \in \{0,1\}, \quad i \in J, k \in I.$$